■ **Emulazione** Fai rivivere il Commodore PET e programmalo!

LINUX PRO

L'unica guida libera al mondo dell'Open Source

NEL WEB DVD le istruzioni a pagina 4

- GhostBSD 21.09.08
- MX-Linux 21 beta 2
- Art Of Illusion 3.2.0
- Dust3D
- Rescuezilla 2.2
- Whonix XFCE
- Haguichi 1.4.4
- Media Downloader 1.6.0
- *e molto altro...*

FAI STREAMING A COSTO ZERO

Manda la tua vita in diretta sul Web

- Organizzi le dirette gratis con OBS Studio
- Trasmetti da dove vuoi con una Pi e una Webcam
- Con Owncast puoi abbandonare YouTube e Twitch

8,6 GB DI SISTEMI OPERATIVI E APP!

GhostBSD: elegante, semplice e basato su BSD

Salva i dati!

RIPRISTINA SUBITO QUALSIASI SISTEMA

Fai resuscitare un disco, recupera i documenti e proteggi le partizioni

NUOVO! 2 DVD DA SCARICARE CON IL CODICE ALL'INTERNO IN ESCLUSIVA PER TE

Blinda le tue Web app

Scopri i bug delle tue applicazioni per Internet ed elimina le vulnerabilità

Bimestrale - N.209 - 5,90 €

P.I. 08-10-2021 Ottobre-Novembre

Modellazione
3D da professionisti
» Dai vita a incredibili progetti con i software gratuiti più potenti

Gestione file
Crea il tuo NAS con Ubuntu
» Sfrutta Cockpit e Docker per un potente server a costo zero

TARIFFA R.O.C. - POSTE ITALIANE SPA - ABB. POST. - DL. 353/2003 (CONV. IN L. 27/02/2004 N. 46) ART. 1, COMMA 1, S/NA - DISTRIBUTORE: PRESS-DI, SEGRATE

15 minuti di fama grazie a Linux

"In futuro, tutti saranno famosi a livello mondiale per 15 minuti": così dice una famosa citazione (non di Warhol, come spesso si dice) nata alla fine degli anni Sessanta e che spiega perfettamente il mondo attuale mosso principalmente da interazioni sui social su Internet. Eppure non è detto che questo concetto che spesso vediamo come negativo sia così male: ci sono tante persone che con un po' di visibilità potrebbero comunicare in maniera molto positiva con tutto il mondo. Chi, per esempio? Beh, tu. Sì, tu che stai leggendo questo numero di Linux Pro e magari sai come programmare script Python perfetti per controllare dispositivi a distanza. Oppure parleresti per ore di distribuzioni oscure. Bene, perché non iniziare oggi a parlare col resto del mondo con dirette video perfettamente prodotte? Ecco quindi che arriva in soccorso l'articolo di copertina di questo numero, che mostra come per diventare veri Twitcher o YouTuber non serve spendere nulla. Non solo: puoi usare la tua piattaforma grazie a Owncast! Questo poi è un numero ricchissimo di contenuti per tutti i gusti: dai trucchi per creare perfetti script asincroni a modellare oggetti tridimensionali con software di livello professionale, abbiamo messo insieme davvero il meglio per chiunque sia appassionato di Open Source. Allora buon divertimento e... facci sapere se apri il tuo canale video!

La redazione di Linux Pro

Da non perdere su questo numero...

QUESTA CARTA RISPETTA L'AMBIENTE

CONTATTI

Domande alla redazione: **redazione@linuxpro.it**
Abbonamenti: **abbonamenti@linuxpro.it**
Arretrati: **arretrati@linuxpro.it**
Problemi con il DVD: **aiutocd@sprea.it**
Sito Web: **www.linuxpro.it**
Oppure inviate le vostre lettere a:
***Linux Pro*, Sprea S.p.A.,**
Via Torino 51, 20063 Cernusco S/N
Telefono: 02.92432.1

@Italia_news

LINUX PRO 209

Sommario LINUX PRO

Benvenuti nel duecentonono numero di Linux Pro, la guida definitiva a Linux e al mondo Open Source

Cover story

FAI STREAMING A COSTO ZERO

ABBONATI ALLA
VERSIONE DIGITALE
SOLO PER PC E MAC
A SOLI 14,90 €
DURATA ABBONAMENTO 1 ANNO
www.sprea.it/digital

Sommario

IL PROSSIMO NUMERO IN EDICOLA DAL 10 DICEMBRE

IL DVD IN BREVE

DVD A

DISTRIBUZIONI

- GhostBSD 21.09.08
- MX-Linux 21 beta 2

DVD B

DISTRIBUZIONI

- Rescatux 0.73
- Rescuezilla 2.2
- SystemRescue 8.04
- Whonix XFCE 16.0.2.7

RIVISTA

- Art Of Illusion 3.2.0
- Dust3D
- Haguichi 1.4.4
- Media downloader 1.6.0
- Audacity 3.0.4

E altro ancora!

Prova la tua rivista anche in digitale

www.linuxpro.it/abbonamenti

Guida DVD

Ogni mese Linux Pro vi offre i programmi e le distribuzioni più recenti su DVD

Sul DVD di questo mese...

Giochi e strumenti da installare subito!

» GhostBSD 21.09.08
» Art Of Illusion 3.2.0

Le migliori distro

» Rescatux 0.73
» Rescuezilla 2.2
» SystemRescue 8.04
» Whonix XFCE 16.0.2.7
» MX-Linux 21 beta 2

I migliori programmi selezionati p. 50

» Dust3D
» Haguichi 1.4.4
» Media downloader 1.6.0
» Audacity 3.0.4
» Dragit 0.6.1
» Radiotray-NG 0.2.8

Distro

GhostBSD

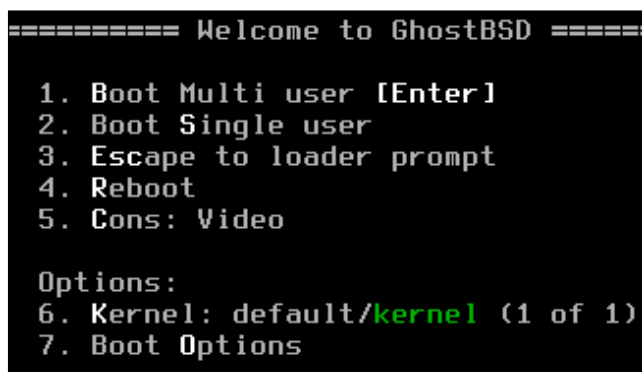
Basato su **FreeBSD** e **MATE**, con ambiente desktop **GTK**, questo sistema operativo punta molto sull'estetica e sulla leggerezza per conquistare la propria fetta di mercato. L'acronimo **BSD** sta per **Berkeley Software Distribution**, poiché le sue radici vanno trovate proprio nella **Berkeley Unix Research** della Università della California. L'attuale versione disponibile di GhostBSD risale allo scorso 8 settembre, quindi è piuttosto recente e contiene le ultime correzioni ai **bug** più importanti.

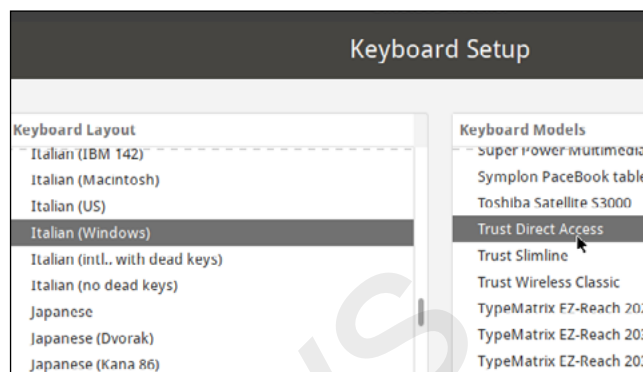
Interfaccia amichevole

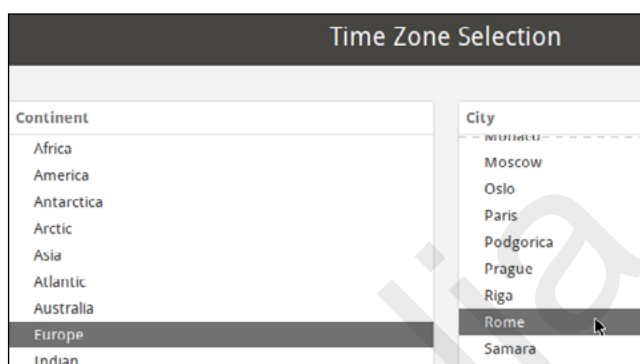
Chi è già abituato a muoversi in sistemi operativi basati su MATE non avrà la benché minima difficoltà a orientarsi sin dal primo avvio di GhostBSD. Tuttavia anche i neofiti di **Linux** ci metteranno pochissimo a muovere i primi passi. Tre sono le icone presenti sul desktop: **Computer**, che fa accedere al file manager, **Home di** (nome utente), che apre quella specifica cartella, e **Cestino**. In basso a sinistra c'è il pulsante che permette di minimizzare o di riaprire con un click tutte le finestre attive. Invece, in basso a destra, ci sono i quattro ambienti di lavoro disponibili, anch'essi visualizzabili con un click. È quindi nella barra superiore che si trovano i menu che permetteranno all'utente di usare il sistema operativo e le sue risorse.
In **Applicazioni** viene visualizzato un menu con sette voci tematiche, nelle quali sono raggruppati i vari programmi preinstallati e in cui finiranno quelli che verranno installati successivamente. In totale ci sono trenta tra programmi veri e propri e semplici accessori.
Il menu **Risorse** è dedicato alla visualizzazione delle cartelle principali, alla gestione della rete domestica o aziendale e del server. Invece nel menu **Sistema** troverete tutti quegli strumenti che permettono la personalizzazione di GhostBSD. Per esempio, nel sottomenu **Preferenze**, è possibile modificarne l'aspetto, mentre il sottomenu **Hardware** è dedicato alle periferiche. Sempre in Sistema, nel sottomenu **Personale**, c'è lo strumento per gestire le applicazioni d'avvio. Un altro strumento importantissimo si trova nel sottomenu **Amministrazione** ed è **Update Station**, che serve per aggiornare il sistema operativo. Infine in Sistema ci sono anche i comandi per riavviare o spegnere il computer, oppure cambiare utente. Sul lato destro della barra superiore c'è il controllo del volume audio e del collegamento a Internet, oltre al calendario. Se volete arricchire GhostBSD con ulteriori applicazioni, avrete a disposizione **Software Station**, che si avvia da **Sistema>Amministrazione**. Sulla sinistra c'è il pannello con le numerosissime aree tematiche, come **games** o **multimedia**. Selezionata una di queste, sulla destra appare l'elenco dei programmi disponibili. Dopodiché basterà sceglierne uno, fare click su **Apply** e poi su **Confirm**, nella finestra che si apre. Quindi niente di più semplice e pratico per avere sempre un sistema operativo pronto a rispondere alle esigenze dell'utente.

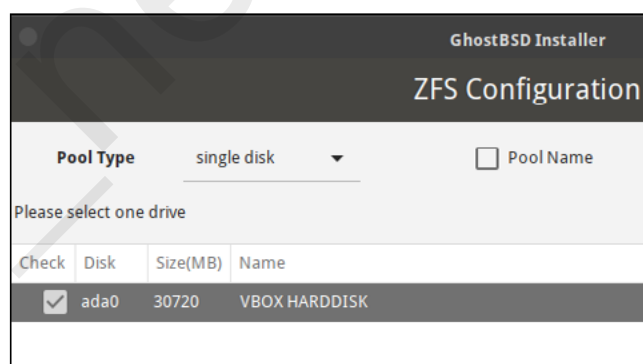
SCARICA SUBITO I DUE DVD COMPLETI INSERENDO IL CODICE HZJDIRD

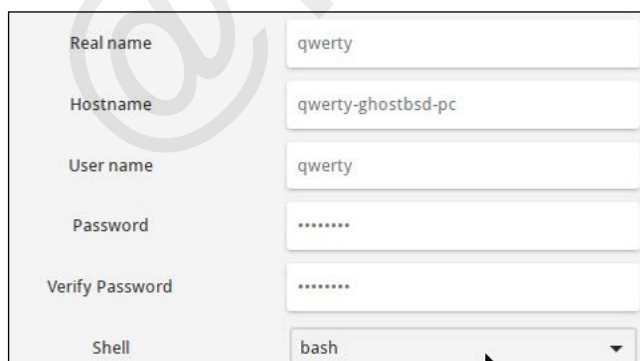
@italia_news

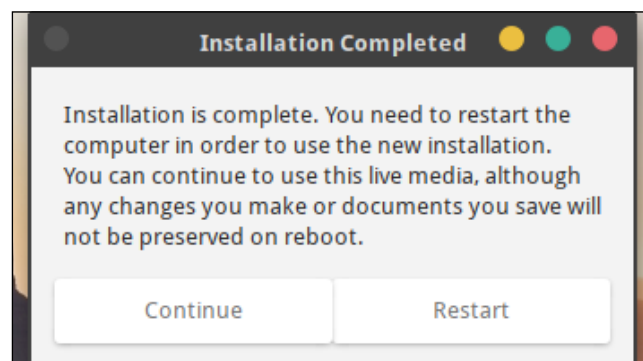
Dalla versione Live all'installazione definitiva

1 La prima scelta

Quando avviate il file **ISO**, viene visualizzato un menu con diverse opzioni. Per esempio, se premete semplicemente **INVIO** il sistema operativo viene caricato in modalità **Live**. Per installarlo in maniera stabile e definitiva, basta fare un doppio click sull'icona **Install GhostBSD**, presente sul desktop.

2 Lingua e tastiera

Nel pannello a sinistra della schermata **WelcomeTo GhostBSD!** selezionate **Italiano**, dopodiché fate click su **Next** per proseguire. In **Keyboard Setup**, a sinistra scegliete il tipo di tastiera che avete, per esempio **Italian (Windows)**, e a destra la marca. Quindi premete su **Next**.

3 Fuso orario e il tipo di installazione

A sinistra della schermata **Time Zone Selection**, selezionate **Europe** e a destra **Rome**. Poi fate click su **Next**. Nella schermata successiva, **Installation Type**, selezionate una delle due opzioni. La più semplice è **Full disk configuration**, che userà l'intero disco rigido del vostro computer. Quindi premete su **Next**.

4 Disco rigido e opzione di boot

In **ZFS Configuration**, selezionate il disco rigido su cui volete installare il sistema operativo e fate click su **Next**, poi ancora su **Next** nella schermata seguente, dopo avere selezionato l'opzione di **boot**. Quindi digitate nei due campi disponibili la password di amministratore e premete su **Next**.

5 Creazione dell'utente

In **User Setup**, compilate il campo **Real name** con il vostro nome utente, poi digitate una password in **Password** e **Verify Password**, quindi nel menu **Shell** selezionate per esempio **bash**. A questo punto non resta che fare click su **Install** per dare inizio alla procedura di installazione vera e propria.

6 Primo avvio di GhostBSD

Terminata l'installazione, fate click sul pulsante **Restart**, dopo esservi assicurati di avere rimosso il supporto con il file **ISO**, altrimenti tornerete alla versione **Live**. In ogni caso, dovrete premere **INVIO** nella schermata iniziale per accedere al sistema operativo, che vi richiederà la password di amministratore. LXP

FAI STREAMING A COSTO ZERO

Come sfruttare la potenza di OBS Studio per portare il meglio ai vostri follower con uno streaming impeccabile

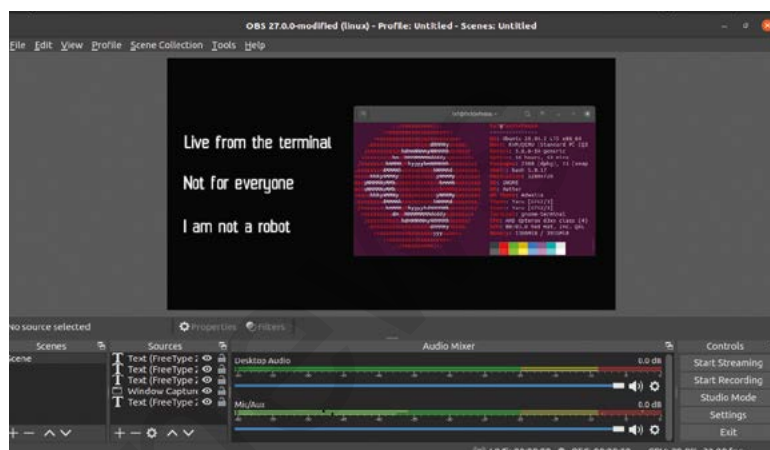
Oggi quella dello streamer online è una professione riconosciuta e pure gli hobbisti che hanno il proprio canale video sui social sono sempre di più, anche a causa della spinta data dalla pandemia.
Che desideriate insegnare a riparare lavatrici su YouTube, a giocare a Doom su Twitch o semplicemente vogliate sfogarvi su Facebook Live, potete fare contenuti di qualità. Organizzare uno studio di registrazione professionale a casa è costoso, ma in realtà non serve per iniziare. Molti YouTuber di grande successo hanno esordito armati solo dei loro laptop.
Dal punto di vista del software, il prodotto Open Source **OBS Studio** è la scelta più popolare, grazie alla sua interfaccia intuitiva che vi permette di manipolare le scene e aggiungere effetti durante la registrazione.
Può trasmettere in **streaming** su tutte le più popolari piattaforme (e molte altre). Imparando a sfruttarlo al meglio potrete partire con il piede giusto alla conquista di Internet. Si può persino usare con una **Raspberry Pi** e una **Raspberry Pi High Quality Camera**. Grazie a **Owncast**, poi, potrete fare il **self-hosting** dei vostri stream con grande facilità. Vi serve solo abbastanza banda da supportare il vostro pubblico e basta una discreta velocità di caricamento per poter gestire qualche decina di **follower**. Il passo successivo è utilizzare il programma su un **VPS** (**Virtual Private Server**). Una volta che il vostro stream passa fluidamente da OBS a Owncast, il trasporto pesante verso gli utenti avviene sul **cloud**.

ABC dello Streaming

L'hardware e l'infrastruttura che vi servono per trasmettere dal vivo

Si sente spesso parlare della natura narcisistica della comunicazione nel ventunesimo secolo e di come il focus sull'individuo, espresso nella cultura dei selfie, dei social media, degli influencer e delle pubblicità targettizzate, sfrutti l'ego delle persone per guidare le loro scelte non solo negli acquisti ma anche nella politica e nella vita. D'altro canto, il lato positivo della popolarizzazione della tecnologia per la comunicazione è che tutti possono avere una voce e si possono sviluppare tanti contenuti di qualità che solo qualche anno fa non avrebbero potuto raggiungere un ampio pubblico. Imparare a sfruttare videocamera, microfono e le magie dello **streaming** è infatti un ottimo modo per poter condividere le proprie conoscenze e il proprio talento, dall'arte alla tecnologia.

L'attrezzatura di base

Qualunque contenuto vogliate condividere, vi serviranno una videocamera e un microfono. Un **laptop** compatibile con **Linux** con installato **OBS** (**Open Broadcasting Software**) **Studio** è un ottimo punto di partenza. Al crescere delle vostre ambizioni e del vostro pubblico, potreste però voler migliorare la vostra attrezzatura. I microfoni di qualità eliminano molti problemi e una buona illuminazione vi permette di venire meglio in video. Se fate lo streaming di videogiochi può servirvi una **scheda di acquisizione video**, se diffondete audio dal vivo potreste investire in un **banco di mixaggio** o in una scheda audio migliore e per inserire fondali di fantasia vi serve uno **sfondo da chroma key**. Prima di lanciarvi in investimenti, però, dovete aver capito bene le basi di come funziona lo streaming. OBS invia gli stream video (che arrivino dal vostro desktop, da una videocamera o da Internet) a una **piattaforma di streaming**. Questa, che può essere **YouTube** quanto un server a casa vostra, manda gli stream agli spettatori, spesso ricomprimendoli **on the fly** per accomodare le diverse bande disponibili. A OBS, quindi, serve solo abbastanza banda in caricamento da poter portare i dati alla piattaforma di streaming, che dovrebbe averne di più. Le piattaforme incorporano in genere anche qualche forma di interattività, come **finestre di chat**, in modo che i vostri spettatori possano esprimere la propria approvazione o dissenso con un testo o tramite **emoji**. Nel seguito dell'articolo esploreremo anche come impostare il proprio server per lo streaming **self-hosted**, ma prima di tutto bisogna familiarizzare con OBS.

Anche se non avete una videocamera, potete fare il **livestreaming** dei vostri **tutorial** da **terminale** con un semplice microfono

» STREAMING E REGISTRAZIONE A CONFRONTO

In questo articolo ci si concentra sul **livestreaming**, ma se preferite registrare i vostri contenuti, in modo da potervi apportare modifiche o rifare delle parti che non vi piacciono, potete poi caricarli. **OBS Studio** è utile tanto per le registrazioni quanto per lo **streaming**: dovete solo avere abbastanza spazio su disco. Al **bitrate** predefinito (**2.500 kbit/s**) un'ora di video occupa poco più di un gigabyte (a cui va aggiunto lo spazio per l'audio). Se registrate su un'altra macchina, per esempio nella vostra **LAN** con **Samba**, verificate di avere abbastanza banda, soprattutto se fate streaming e registrazione insieme. Il **wizard di configurazione** di **OBS** può ottimizzarlo per la registrazione sin dall'inizio. Dopo aver registrato il vostro video, per l'elaborazione potete usare l'**editor video non lineare KDEnlive**. Vi servirà però ancora più spazio, perché lavorare con i video tende a produrre molti **file temporanei** oltre al risultato finale. Potreste anche elaborare l'audio separatamente, per esempio utilizzando **Audacity**.

Elaborare l'audio separatamente dal video vi permette di ottenere risultati più professionali e **Audacity** è un ottimo strumento per farlo

Usare OBS Studio

Il miglior software per l'home streaming è Open Source, potente e molto facile da imparare: dovete solo cominciare!

Provate a usare OBS con il computer prima che con la Raspberry Pi. Probabilmente è già presente nei repository della vostra distribuzione, ma vale sempre la pena di scaricare l'ultima versione (che è la 27.0.1 al momento di andare in stampa). Il programma ha il suo repository per Ubuntu e trovate le istruzioni relative all'indirizzo https://bit.ly/38UEg2g, ma per questo articolo viene usato il pacchetto Snap. Include il supporto per la codifica video assistita dall'hardware, per le fotocamere virtuali e molto altro che probabilmente non otterreste compilando OBS direttamente. Non avete bisogno di usare Ubuntu (per il quale lo trovate nel Software Center) per utilizzare Snap e, una volta installato il suo demone, basta il comando:

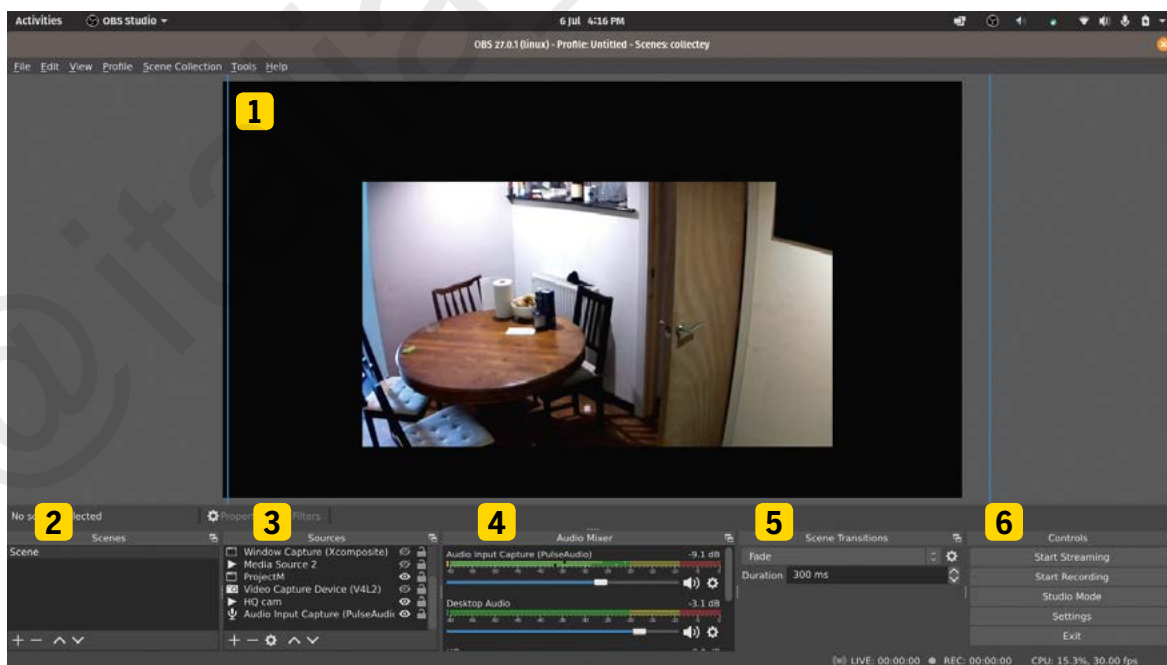
```
$ sudo snap install obs-studio
```

Per poter comunicare con le interfacce e l'hardware corretto, lo **Snap** deve essere connesso come segue (viene fatto automaticamente se lo installate da **Ubuntu Software Center**):

```
$ sudo snap connect obs-studio:{alsa,audio-record,avahi-control,camera,jack1,kernel-module-observe}
```

Se non vi piacciono gli Snap potete usare il **pacchetto Flatpak** disponibile. Lo abbiamo testato con un laptop con **Pop!_OS** senza incontrare problemi. Spesso potrete installarlo direttamente dal software center della vostra distribuzione, ma se così non fosse basta installare il **demone** di **Flatpak** e poi inserire i comandi:

```
$ flatpak install flathub com.obsproject.Studio
$ flatpak run com.obsproject.Studio
```

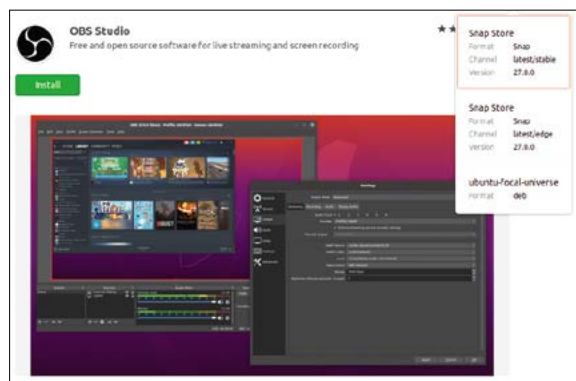
ORIENTARSI NELL'INTERFACCIA DI OBS

1 Finestra di anteprima Mostra la scena (**Scene**) selezionata. La impostate trascinando e ridimensionando le fonti con le loro cornici.

2 Area delle scene Una scena è un **template** per una sezione della vostra trasmissione. Possono usare diverse fonti, **layout** o intestazioni.

3 Selezione delle fonti Possono essere videocamere, microfoni, finestre della scrivania o altro. Ogni scena ha le sue fonti.

4 Controlli dell'audio I **VU meter** reagiscono all'**input audio**. Non devono raggiungere il rosso. Nel caso, usate gli **slider** per abbassare i livelli.

5 Transizioni delle scene Qui potete scegliere un bell'effetto, come una **dissolvenza**, per rendere più gradevole il passaggio tra le scene.

6 Controlli dello stream Avviate o interrompete lo **stream** (o la registrazione) con i pulsanti o passate in **Studio Mode** per inserire degli effetti dal vivo.

La versione **Snap** di **Open Broadcasting Software Studio** include un'ampia serie di "chicche" aggiuntive

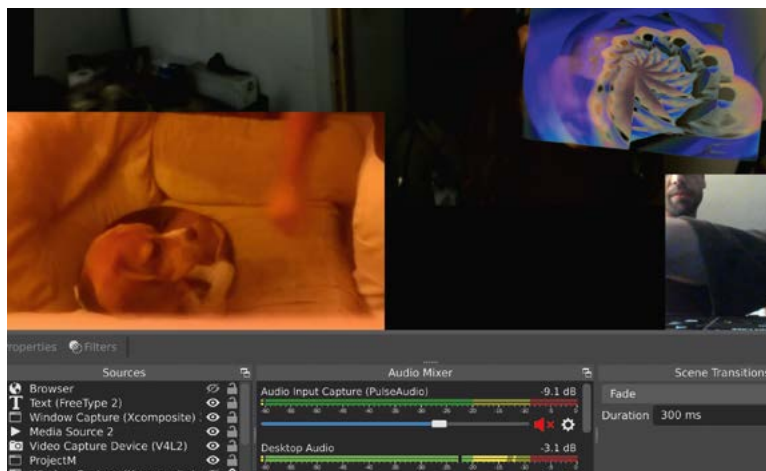
All'avvio di **OBS** parte un **wizard per la configurazione**. Scegliete **Optimize for streaming** e utilizzate la risoluzione predefinita di **1.280 x 720**. È consigliabile ridurre i **fotogrammi al secondo** (**FPS**) a 30, almeno all'inizio. Dovete poi scegliere una **piattaforma di streaming**, sulla quale è necessario avere un **account** e di cui dovrete rispettare termini e condizioni. Viene usata una chiave per verificare la vostra identità. Fate click sul pulsante **Get Stream Key**, fate il **login** nella vostra piattaforma, copiate la **chiave stream** privata dalla pagina Web e incollatela in OBS. Lasciate la spunta a **Estimate bitrate**, che permette al programma di configurare l'**encoder** in base alla vostra banda e alla potenza della **CPU**. La piattaforma distribuirà quindi il vostro stream, offrendo anche bitrate inferiori. Accettate le impostazioni suggerite da OBS (potete sempre cambiarle in seguito) per entrare nella sua interfaccia principale. Se avete collegati videocamera e microfono vengono impostati automaticamente e potete vedere in una finestra di anteprima ciò che inquadra la videocamera.

Primi passi con il vostro hardware

Videocamera e microfono sono mostrati come **Video Capture Device** (**V4L2**) e **Audio Input Capture** (**Pulseaudio**) nella sezione **Sources**. La risoluzione del **canvas** è stata impostata a **720 p** prima, ma può essere diversa da quella della vostra fotocamera. Potete cambiarla facendo click col tasto destro sulla fonte V4L2 e scegliendo **Properties**. Se non volete riempire il canvas con l'**input** della videocamera, scegliete una risoluzione più bassa e in seguito potrete aggiungere delle decorazioni nelle parti vuote. La fonte attualmente selezionata appare circondata da una cornice rossa e potete trascinarla e ridimensionarla con le maniglie anche durante una trasmissione video. Ridimensionare i video è però impegnativo e se volete usare la stessa fonte in dimensioni diverse (per esempio per mettere a schermo intero un videogioco) è più efficiente usare le **Scene**. Potete impostare la fonte a risoluzioni diverse per ognuna di esse. Se volete solo una fonte video, fate click col tasto destro sulla sua anteprima e scegliete **Resize Output (source size)**, il che imposta la risoluzione di base e dell'**output** su quelle della videocamera, facendola apparire a schermo intero. Il video viene inviato alla piattaforma di streaming alla risoluzione dell'output. Se volete fare uno stream solo audio (per esempio per **Mixcloud**) avete una classe di contenuto multimediale in meno di cui preoccuparvi. In ogni caso, assicuratevi che i livelli di registrazione siano buoni e familiarizzate con come si mette in **mute** il microfono per evitare spiacevoli inconvenienti. OBS salva automaticamente le fonti e le scene, quindi è facile riprendere se c'è un'interruzione, ma è comunque utile cambiare il nome della vostra prima scena da Scene a qualcosa di più descrittivo. Lo potete inserire facendo click col tasto destro e selezionando **Rename**. Vale la pena di rinominare anche alcune delle fonti. Anche le piattaforme offrono varie opzioni dalla loro **dashboard**. **Twitch**, per

TUTTO DA SOLO

"Con Estimate bitrate, OBS configura automaticamente l'encoder in base alla vostra banda e alla potenza della CPU"

esempio, vi permette di contrassegnare i contenuti pronti, impostare la **latenza** e salvare le trasmissioni. La sezione **Studio** di **YouTube** consente di modificare i video **online** o aggiungere una colonna sonora. Dal punto di vista dei contenuti, ovviamente non c'è una ricetta per il successo, ma è sempre importante pianificare cosa volete dire e fare. Fare click sul box **Scene Collection** durante la trasmissione dal vivo è facile quindi se avete organizzato bene le vostre scene prima di iniziare non avrete problemi a sfruttarle. Una volta che avete acquisito padronanza della **Studio View**, inoltre, potete anche creare delle transizioni professionali tra di esse.

In poco tempo potrete aggiungere al vostro **livestream** cani e portali interdimensionali...

Streaming con la Pi

Potete utilizzare un vecchio modello come videocamera remota oppure decidere di gestire tutto con una Pi 4

Non sarebbe comodo (ed economico!) per i vostri progetti poter utilizzare la Raspberry Pi, magari con la sua videocamera ad alta qualità? Abbiamo provato per voi delle possibili configurazioni con hardware più e meno recente e potente.

La **HQ Camera** ha un supporto per **treppiede** standard, molto utile perché i cavi tendono a sbilanciarla

Abbiamo fatto la prima prova collegando a una **Pi Zero** una **Pi Camera** standard. Va sottolineato che, se configurate una Pi **headless** da zero, il servizio **SSH** è disabilitato **di default** e dovete comunicarle anche se lavorate con un **collegamento wireless**. Per farlo mettete un file vuoto di nome **ssh** e uno con le informazioni sulla rete wireless (**wpa_supplicant.conf**) nella partizione di avvio della **scheda SD** della Pi. Eseguitevi quindi `raspi-config` per abilitare la fotocamera nella sezione **Interfaces**. I programmi **raspistill** e **raspivid** inclusi con la Pi sono comunemente usati per testare il funzionamento della videocamera, ma potete utilizzare il secondo anche per convertire la vostra Pi in una fonte in rete in **OBS**. Collegatevi via SSH ed eseguite il comando:

```
$ raspivid -t 0 -l -n -w 640 -h 360 -o tcp://<Indirizzo IP>:3333
```

sostituendo a `<Indirizzo IP>` quello della vostra Pi. Ci sono moltissime opzioni per il programma, ma quelle che vi servono gli indicano (nell'ordine) di continuare lo streaming indefinitamente, mettersi in ascolto su un **socket TCP**, non mostrare l'anteprima e inviare uno stream **640 x 360** alla **porta 3333** dell'interfaccia indicata. Potete specificare tutte le interfacce con **tcp://0.0.0.0:3333** ed è anche possibile fare lo streaming via **UDP**. Per usarlo in OBS, aggiungete una fonte multimediale, togliete la spunta al riquadro del file locale e usate lo stesso URL **tcp://<Indirizzo IP>:3333** di prima. La fonte appena aggiunta sarà vuota per qualche secondo, poi vedrete uno stream video. Se usate un modello

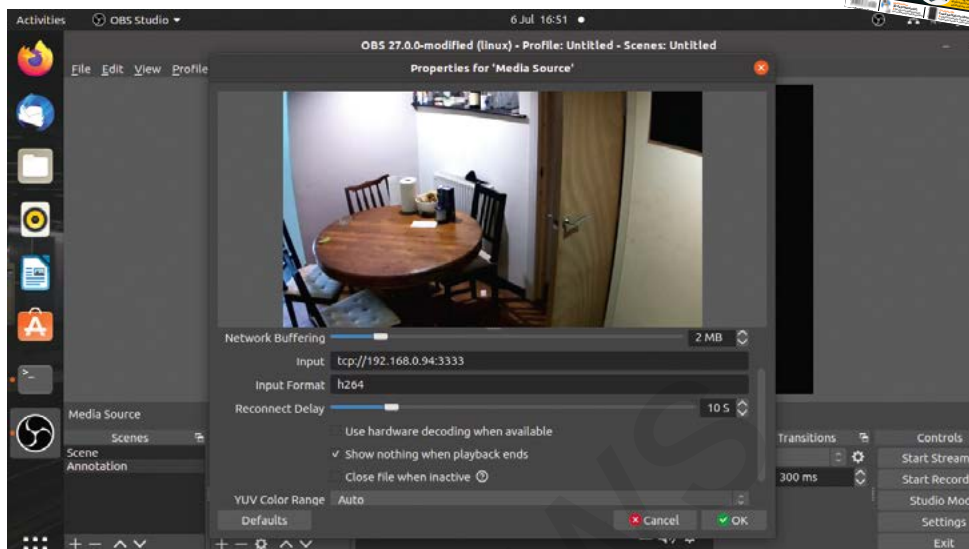
» UN APPROCCIO "FAI DA TE" ALLA PI CAMERA

Nessuna delle due **Pi Camera** ha un microfono incorporato, quindi non basta se volete dell'audio nei vostri **stream**. **Dell** ha recentemente spostato la **webcam** dei suoi **laptop XPS13** al centro, ma se avete un vecchio modello scentrato o comunque la webcam integrata non vi soddisfa, potete sfruttare la Pi Camera come surrogato insieme al microfono del laptop. Se adottare il metodo con `raspivid` sulla **Pi 4**, la **latenza** viene sicuramente ridotta, ma non azzerata. Potete però compensare in **OBS** facendo click sull'ingranaggio accanto agli indicatori **Audio Input Capture**, entrando in **Advanced Audio Properties** e inserendo un ritardo sfruttando il riquadro **Sync Offset**. Ottenere le impostazioni giuste non è molto facile, perché la latenza effettiva può variare in modo significativo in base a quanto può essere compressa la scena corrente e alle condizioni della rete. Non tutto è perduto, però. Grazie alle porte **OTG** della **Pi Zero** e della Pi 4, infatti, potete impostare questi modelli come **webcam USB**. Un **post** del **blog** di **David Hunt** all'indirizzo **https://bit.ly/2Vyb6Dm** offre tutti i dettagli per farlo (in inglese, come i successivi). Se invece volete un approccio più automatizzato potete visitare il **GitHub** di **Jeff Geerling** all'indirizzo **https://github.com/geerlingguy/pi-webcam**. Un altro metodo è utilizzare **Ubuntu Core** per fare lo **streaming** con **MJPEG** e trovate un ottimo post sul blog di **Canonical** all'indirizzo **https://bit.ly/3E0ILH5**.

come la Pi Zero il ritardo è significativo perché il **system on a chip** deve lavorare un po' per codificare lo stream nel formato **h.264**, particolarmente se avete una rete wireless, quindi può non essere la soluzione ideale per stream articolati. Siamo quindi passati a testare una **Raspberry Pi 4** da 8 GB con una **High Quality Pi Camera** e l'**obiettivo grandangolare** da **6 mm.** Ce ne è anche un altro ufficiale, il **teleobbiettivo**, ma è meno adatto per lo streaming, a meno che non dobbiate riprendere oggetti lontani. Questa configurazione è stata facile da realizzare ma se usate l'obiettivo da 6 mm ricordatevi di rimuovere l'**adattatore CS-mount** del modulo fotografico o non riuscirete a mettere a fuoco.

Media Sources offre ogni genere di opzione, ma non siamo riusciti a far attivare l'**accelerazione hardware**

Restate focalizzati

L'obiettivo ha due ghiere che servono per regolare **messa a fuoco** e **diaframma** e che potete bloccare con il cacciavite incluso. Se non siete abituati a usare gli obiettivi manuali dovrete probabilmente fare un po' di pratica, ma potete impostare **raspivid** come visto in precedenza per verificare che state andando nella direzione giusta. Un buon metodo è puntare la videocamera nel punto che volete riprendere, poi regolare il diaframma fino a ottenere una buona illuminazione, fissare con il cacciavite la ghiera dell'apertura e infine impostare la messa a fuoco. A questo proposito, ricordatevi che la messa a fuoco si può regolare anche sulla Pi Camera standard. Nei vecchi modelli dovete avere il coraggio di togliere la colla attorno alla lente, ma poi potete ruotarla fino ad avere una regolazione vicinissima per immagini macro. **Pimoroni** vende uno strumento per farlo, oltre a vari obiettivi per entrambi i modelli di videocamera. Per completare le nostre prove sul campo, abbiamo deciso di fare qualcosa di più avventuroso. Non c'è una **build** ufficiale di **OBS** per la Pi, ma è un programma Open Source quindi abbiamo deciso di compilarlo direttamente. Potete farlo anche con una **Pi 3**, ma vi servirà un **file di swap** perché **1 GB** di **RAM** non basta e i tempi saranno lunghi. C'è un **thread** su questo tema su un **forum** del programma (**https://bit.ly/3ninUcw**) a cui potete fare riferimento se siete in difficoltà. Ci possono essere un po' di problemi perché degli aggiornamenti sia di Raspberry Pi OS sia del codice di OBS interferiscono. Prima di iniziare dovete inoltre installare numerosi pacchetti e sarebbe inutile inserire qui un lungo elenco. Potete invece usare lo **script** di **Xavier Alonso** che trovate all'indirizzo **https://github.com/xbelanch/OBS4Pi**. Dovete fare prima l'aggiornamento alla versione **Bullseye** di **Pi OS**. A questo scopo aprite **/etc/sources.list** e cambiate tutti i riferimenti a `buster` in `bullseye`. Fate l'**update** e l'**upgrade** con **apt** e riavviate. Scaricate quindi il file **build.sh** da **GitHub**, rendetelo eseguibile e lanciatelo con:

```
$ wget https://github.com/xbelanch/OBS4Pi/raw/main/build.sh
$ chmod 755 build.sh
$ ./build.sh
```

La Pi impiegherà molto tempo per l'operazione e potrebbe surriscaldarsi parecchio. Oltre a OBS, lo script compila versioni **custom** di **FFMPEG** (l'edizione standard non supporta il protocollo **Secure Reliable Transport** o **SRT**) e tutti i **codec** associati. Se tutto viene completato senza errori, potete avviare OBS da riga di comando. Non è stato scaricato il **launcher** dal GitHub di Alonso, quindi le istruzioni per usarlo non funzionano. Se però provate a eseguire direttamente il comando `obs`, compare un messaggio di errore su **contesti OpenGL** non validi, perché OBS non conosce le capacità video della Pi. Per superare questo inconveniente, forzate la versione di OpenGL con il comando:

```
$ MESA_GL_VERSION_OVERRIDE=3.3 obs
```

oppure scaricate il launcher di Alonso. A questo punto potete fare streaming con la Pi... e iniziare a sperimentare con mille accessori!

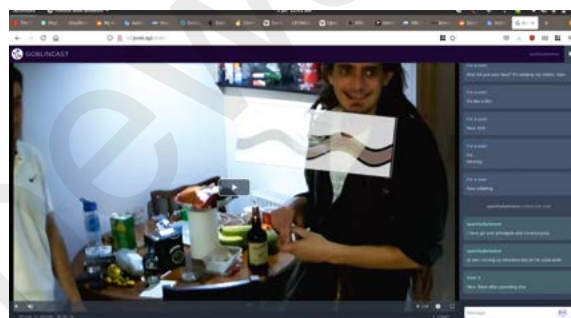
COMPILARE OBS PER LA PI

"Non c'è una build ufficiale di OBS per la Pi, ma è un programma Open Source quindi abbiamo deciso di compilarlo direttamente"

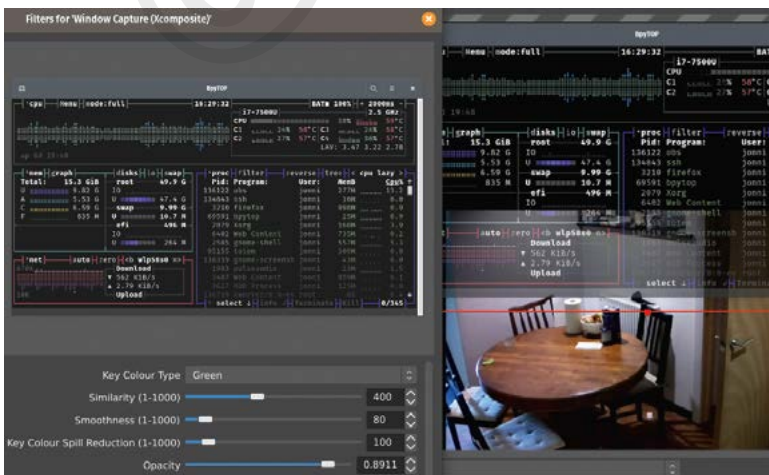
Più controllo sui vostri contenuti e utenti

Scoprite cos'altro può fare OBS e poi liberatevi delle corporation e fate il self-hosting dei vostri stream con Owncast

Ora che avete chiare le basi della distribuzione di contenuti con lo streaming può essere utile esplorare cos'altro può offrirvi OBS. Abbiamo già accennato al fatto che si possono creare meraviglie con un po' di organizzazione, il riquadro Scene Collection e la vista Studio View durante le trasmissioni dal vivo. Nella pratica, facendo click sul pulsante **Studio View** vedete due pannelli. **Preview**, sulla sinistra, mostra la scena che state pianificando e **Program** sulla destra visualizza il contenuto del vostro **streaming** corrente (se lo state facendo). Potete accedere ad alcune semplici transizioni (**Cut**, **Fade** e **Fade to Black**) facendo click sull'icona dell'ingranaggio accanto al pulsante **Transition**. Dopo averle aggiunte, ridisponete le fonti nella finestra di anteprima e usate lo **slider** per sovrapporre la nuova scena manualmente oppure il pulsante per farlo automaticamente. Le fonti a cui potete accedere sono di diversi tipi e potete aggiungerle facendo click sul simbolo **+** nel pannello **Sources**. La più semplice è **Text** (**Freetype2**), che vi permette di aggiungere testi ai vostri video. Sceglietela, selezionate **Create New**, date alla fonte un nome identificativo, scrivete il testo e formattatelo come preferite. La modalità **Chat log** permette di mostrare i più recenti **log IRC** (o di altro tipo), ma se la vostra **piattaforma di streaming** ha la propria chat potete aggiungerla con View > Docks > Custom Browser Docks. Si può anche, in alternativa, adottare Owncast (vedi oltre) che ha la propria interfaccia per le chat. È anche possibile catturare la scrivania o una finestra specifica e usarla come fonte in OBS. Per lo schermo intero usate la fonte Screen Capture (XSHM). Per catturare una finestra, utilizzate Window Capture (Xcomposite). Rimuovete le fonti che non vi servono con il pulsante **-** e usate Scene Collection per riunire le fonti in capitoli. Grazie all'architettura basata su plug-in di OBS è anche facile introdurre funzioni aggiuntive. Se volete arricchire i vostri stream sui videogame cercate con il vostro motore preferito "OBS overlays" sul Web per trovare ogni genere di bordo creativo per le vostre fonti. Può venire naturale pensare che tutte queste fonti siano molto basate su X e non funzionino in Wayland, ma con l'ultima release del programma non dovete preoccuparvi di questo aspetto.

Aggiungere una finestra trasparente di **BPYTop** al vostro stream non è la miglior soluzione per il monitoraggio, ma l'effetto è bello

È molto facile iniziare a lavorare con **Owncast**: non avete bisogno di conoscenze specialistiche o hardware sofisticato

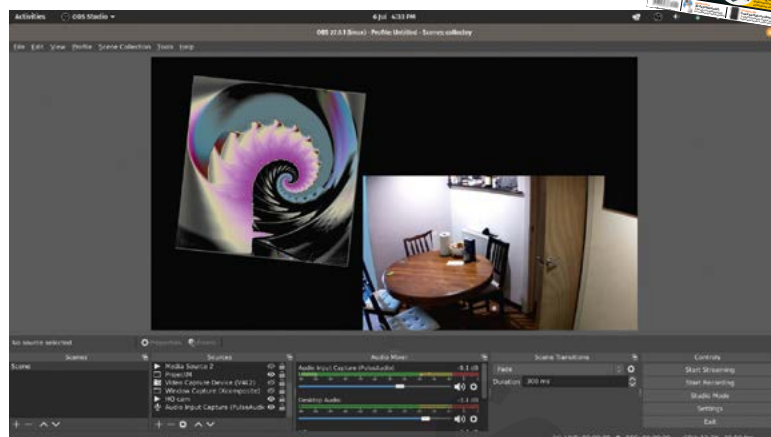
Fate l'hosting dei vostri stream

Le **piattaforme di streaming** proprietarie, con le loro ricche bande e un pubblico sterminato, hanno i loro punti di forza, ma il **self-hosting** vi permette di liberarvi dell'ingerenza dei colossi. Se volete raggiungere l'indipendenza in questo senso, una soluzione efficiente e facile da usare è **Owncast**. La demo sul sito del progetto (**https://owncast.online**) vi mostra come partire in meno di un minuto ma è in inglese e qualche istruzione in italiano può

essere utile. Owncast ha il proprio **installer** online, che si può invocare con il comando:

```
$ curl -s https://owncast.online/install.sh | bash
```

Lo **script** installa anche **FFMPEG** se non lo avete già, perché è necessario per il lavoro di **transcodifica**. Potete poi lanciare Owncast con:

```
$ cd owncast
$ ./owncast
```

Puntate quindi il vostro browser all'indirizzo **http://localhost:8080** (oppure all'**hostname** o all'**indirizzo IP** della macchina remota su cui lo avete installato). Compare l'interfaccia del programma, completa di **chat**. La prima cosa da fare è cambiare la **chiave di streaming** (preimpostata come `abc123`), altrimenti chiunque sulla rete locale, o su Internet se la macchina è accessibile al Web, potrà prendere il controllo del vostro canale. Andate nella pagina di amministrazione su **http://localhost:8080/admin** e sostituite la **stream key** (che è anche la password) nella sezione **Server Settings**. Potete anche cambiare nome, logo e descrizione del sito. Per farlo comunicare con OBS, andate nelle impostazioni di quest'ultimo e, nella sezione **Stream**, configurate il servizio su **Custom**. Come **server** usate **rtmp://localhost/live** (sostituendo **localhost** con la macchina su cui eseguite Owncast) e utilizzate la **chiave stream** impostata prima. Ora se iniziate lo streaming in OBS e andate nella pagina Owncast potete vedervi e farvi un'idea della **latenza**. Quest'ultima si può configurare nel pannello delle impostazioni di Owncast, ma non fatevi tentare dall'idea di abbassarla a valori in millisecondi perché, a e meno che non abbiate una connessione con bassissima latenza con il vostro pubblico, causerete del **buffering**. Nella maggior parte dei casi non è un problema che i vostri fan vedano quello che fate con qualche secondo di ritardo. Perché possano connettersi al vostro stream, la **porta 8080** della macchina che ha in esecuzione Owncast deve essere aperta, così come la **1935** a cui deve collegarsi OBS con il protocollo **RTMP**, quindi dovete modificare le impostazioni del vostro **router**. Ricordatevi però che avete in esecuzione Owncast come utenti, quindi non potete rivendicare

la **porta 80**. Se volete usare il protocollo **HTTPS**, che probabilmente sarà la vostra scelta per qualsiasi utilizzo diverso dalla fase di **testing**, la situazione diventa un po' più complessa. Il metodo raccomandato è quello di usare un **reverse proxy**. Potreste già averne uno configurato, altrimenti trovate le istruzioni per farlo in **Apache** e **Nginx** all'indirizzo **https://owncast.online/docs/sslproxies**. È anche possibile utilizzare lo strumento di amministrazione del server **Site.js** per installare Owncast come servizio con dei semplici comandi da terminale. Il numero di utenti che potete raggiungere con i vostri stream dipende dalla vostra banda e dalla risoluzione che sceglierete. Nei test fatti una risoluzione di **1.500 kbit/s** (**kilobits al secondo**) è risultata adatta ai nostri scopi e, dato che abbiamo una velocità in caricamento teorica di **20 Mbs**, permette di gestire una dozzina di utenti. Si tratta di un numero più che adeguato per fare delle prove o comunicare con qualche amico, ma se volete rivolgervi a un pubblico più ampio potete installare Owncast su un **VPS** e farvi collegare OBS in remoto. Owncast supporta anche i **provider** di **Object Storage** quindi, in cambio di un piccolo investimento, potete far fare il carico pesante del lavoro a fornitori come **Aruba**, **Amazon S3**, **Backblaze** o **Digital Ocean Spaces**. LXP

Con **ProjectM** potete creare delle belle grafiche (in stile **Winamp**) per il vostro audio

» GESTIRE LE QUESTIONI DI COPYRIGHT

Chi si propone di fare il proprio **streaming** con dei contenuti legati al mondo della musica può essere tentato dall'opzione del **self-hosting** perché risolve il problema delle interruzioni del servizio automatiche legate alla gestione dei diritti digitali. Molti utenti di **Twitch** hanno visto bloccare i loro **stream** perché avevano qualcosa che ricordava un brano di musica coperto da **copyright** in sottofondo. Potete naturalmente contestare questi interventi, ma anche nel caso vi venga riconosciuta la ragione il vostro servizio soffrirà dell'interruzione. Se i vostri streaming sono diretti a un ristretto gruppo di amici e l'accesso è protetto da password è improbabile che vi troviate ad affrontare grane legali, ma se le vostre trasmissioni sono pubbliche, magari promosse sui **social media**, e contengono brani musicali protetti da copyright state violando le normative e potete incorrere in sanzioni. Il tema della riproduzione in pubblico e delle licenze **SIAE** è complesso e anche con la consulenza di uno studio legale potete avere problemi con i blocchi automatici (vedi **https://bit.ly/38Q6n2S**) ma trovate tutti i dettagli sulla **Licenza Multimediale Streaming** della SIAE e anche farvi fare un preventivo, se vi serve, all'indirizzo **https://bit.ly/3jUE0qA**.

WEB APP SICURE

La sicurezza applicativa parte dall'adozione di buone pratiche nella pipeline di integrazione e distribuzione continua

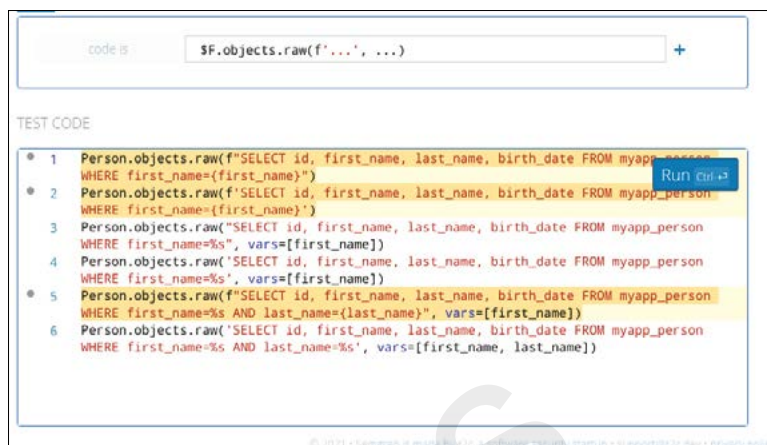
Ogni giorno qualche azienda viene violata dagli hacker e i dati dei suoi clienti vengono sottratti. Tutte dicono sempre che la loro sicurezza era allo stato dell'arte, ma dopo l'evento si finisce spesso per scoprire che un crawler ha trovato e sfruttato un endpoint non protetto.
Molti specialisti informatici hanno sperimentato che le aziende non hanno mai tempo per investire nella sicurezza prima che si verifichi un attacco. Dopo, però, tutti sono molto stupiti e indignati che sia potuto succedere e se la prendono con lo stesso tecnico che da mesi richiede invano i mezzi per migliorare la sicurezza del codice. Per quanto sia triste, non è del tutto colpa dei manager. Chi investe in una nuova funzione vuole vederne i benefici, saperne i tempi di sviluppo, quantificarla.
La sicurezza e il **refactoring**, d'altro canto, sono un processo continuativo che risolve un problema non ancora occorso. Solo riuscendo a far capire costi e benefici dell'investimento in sicurezza senza sembrare allarmisti si può cambiare la cultura dei clienti o aziendale. In questo articolo troverete alcune delle motivazioni e degli strumenti più importanti per iniziare questo percorso che porta anche a produrre software migliore.

Illustrazione: Kym Winters

Le violazioni della sicurezza possono avere effetti devastanti. Se a un'azienda vengono sottratti i dati delle carte di credito dei suoi clienti, quasi sicuramente questi ultimi riusciranno a riavere il denaro sottratto dall'hacker con questa informazione. Se i dati rubati, però, includono quelli bancari, i numeri dei documenti, l'indirizzo e il nome da signorina della madre dell'utente, si può arrivare al furto dell'identità, con conseguenze gravissime.
Proprio per questo un'azienda a cui vengono sottratti i dati può trovarsi a pagare cifre molto ingenti di danni ai clienti coinvolti i dati e, a meno che non sia un colosso, può facilmente finire con il chiudere i battenti per questo motivo.

Partire dall'analisi statica del codice

Il primo passo per aumentare la sicurezza delle applicazioni dovrebbe essere l'**analisi statica del codice**, ossia il processo di valutarlo, senza eseguirlo, sulla base di una serie di regole.
Può sembrare curioso che queste ultime, che sono relative a elementi come lo stile e la formattazione, possano proteggere da vari **attacchi di iniezione di codice**, **denial of service** e altre vulnerabilità.
L'analisi statica non si limita però a questo e per scoprire problemi come **SQL injection**, credenziali integrate nel codice ed **eccezioni non gestite** ci sono ottimi strumenti Open Source curati da alcuni dei migliori esperti di sicurezza al mondo. Usarli è come avere un **team** di persone che vi aiuta a rivedere il vostro codice, evidenziando rapidamente errori o vulnerabilità che potreste aver introdotto. Controllano inoltre se le vostre librerie sono aggiornate e che non abbiate utilizzato nella vostra **app** codice già riconosciuto come vulnerabile. Ci sono vari strumenti per l'analisi statica, alcuni dei quali probabilmente conoscerete già (dei comuni esempi per **Python** includono **Black**, **Dependabot** e **PyLint**) mentre altri sono meno noti (per lo stesso linguaggio, ci sono per esempio **Prospecter**, **McCabe**, **Dodgy**, **Bandit** e **Horusec**).
Si possono sommariamente dividere per categorie: **lint**, analizzatori della complessità, analizzatori di dipendenze/**container** e analizzatori semantici.

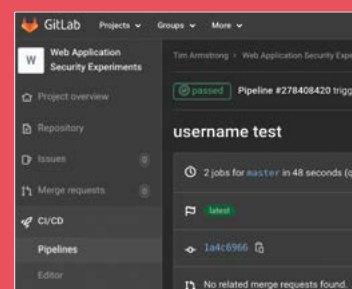
Fate un giro di prova con gli strumenti di **Semgrep**!

Alcuni strumenti, però, hanno caratteristiche di più categorie, come **Semgrep**, un **tool** Open Source che supporta molti linguaggi ed è pensato per permettervi di creare facilmente le vostre regole, sostanzialmente con la stessa sintassi del linguaggio per cui le state scrivendo. Questo vi consente di scrivere semplicemente, per esempio, una regola che identifica l'uso di **f-string** in **statement SQL** (per una **query Object.raw** in **Django** sarebbe `$F.objects.raw(f'...', ...)`) e regole come questa possono essere molto potenti.

Tra bug e pelucchi dei vestiti

I **linter** o lint prendono nome dal primo programma del genere scritto da **Stephen C. Johnson** per il **C** oltre 42 anni fa. Si chiamava Lint dal termine inglese che indica i "pelucchi" che si formano sui vestiti di cotone e lana e che tendono a intasare i filtri della lavatrice. Lo scopo del programma era aiutare l'utente a eliminare elementi e complessità inutili dai programmi, prevenendo i **bug**. Come effetto collaterale si ottiene del codice più pulito e facile da leggere. Le regole di base incluse nei linter riguardano la conformità di stile e formattazione, le variabili non dichiarate e quelle non utilizzate, il codice non raggiungibile, gli errori di sintassi e le eccezioni non rilevate. Oggi la maggior parte degli **IDE** ha linting integrato, ma fare affidamento solo su di esso può portare ad alcune problematiche:

» UN OTTIMO STRUMENTO OPEN SOURCE

La soluzione per l'**integrazione continua** di **GitLab** spicca sulla concorrenza. È Open Source, facile da usare e offre 400 minuti gratuiti nella soluzione **hosted**. Consiste di due elementi: **stage** e **job**. I primi sono come **tag** che raggruppano i job in esecuzioni parallele. Gli stage sono eseguiti in sequenza e se un job all'interno di uno di essi fallisce la sua pipeline si blocca.
La configurazione iniziale di **Gitlab CI** può essere molto semplice:

```
``
stages:
    - prebuild
lint:
  stage: prebuild
  image:
    name: "ckleemann/prospector:latest"
    entrypoint: [""]
  before_script:
    - pip install -r requirements.txt
  script:
    - prospector ./src
``
```

Gitlab offre delle utili schede riassuntive della verifica della **pipeline**

Visibilità – se è nella **pipeline di integrazione e distribuzione continua** (**CI/CD**), il vostro revisore deve solo dare un'occhiata per controllare che non abbiate inserito nuovi problemi.
Uso comune – tutto il vostro team usa lo stesso IDE? Se sono diversi, hanno tutte lo stesso linter?
Estensibilità – se il lint è nella pipeline CI/CD il team può ampliarlo con regole comuni.
Nel caso di Python, avete a disposizione una raccolta sempre in crescita di standard di stile e formattazione, scritti da esperti, nei **PEP** (**Python Enhancement Proposals**, **https://www.python.org/dev/peps/**), tra cui quello che risulta più noto è probabilmente **PEP8**. Strumenti come **PyLint** sono sempre aggiornati con gli standard accettati e aiutano a far sì che il codice sia facile da leggere e mantenere. Questo programma non è però completo come **Prospector**, che include non solo PyLint, ma anche **McCabe**, **Dodgy** e molti altri tool in un'**utility** che presenta tutti i risultati in una struttura comune facile da esaminare e seguire.

```
22 $ bandit -r ./src
23 [main] INFO    profile include tests: None
24 [main] INFO    profile exclude tests: None
25 [main] INFO    cli include tests: None
26 [main] INFO    cli exclude tests: None
27 [main] INFO    running on Python 3.7.1
28 Run started:2021-03-29 18:43:41.045220
29 Test results:
30         No issues identified.
31 Code scanned:
32         Total lines of code: 2
33         Total lines skipped (#nosec): 0
34 Run metrics:
35         Total issues (by severity):
36                 Undefined: 0.0
37                 Low: 0.0
38                 Medium: 0.0
```

Bandit è un ottimo strumento per l'**analisi statica del codice** per la sicurezza e identifica credenziali integrate, vulnerabilità per le **injection** e molto altro

» CODICE MIGLIORE E PIÙ SICURO

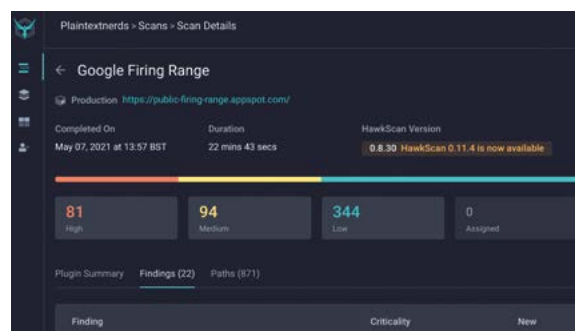
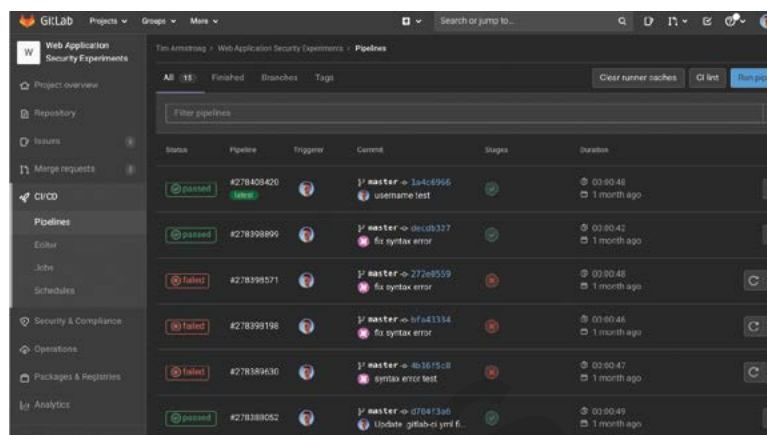
Introdurre l'analisi della sicurezza e del codice nella **pipeline di integrazione e distribuzione continua** (**CI/CD**) è uno dei metodi principali per monitorare la qualità della programmazione e l'esposizione a potenziali rischi. Vi permette di capire (e spesso evitare in modo rapido) alcuni dei **bug** e delle vulnerabilità più comuni. Fortunatamente, aggiungendo un paio di passaggi alla vostra pipeline potete facilmente (e in alcuni casi automaticamente) mitigare intere classi di attacchi, rendendo anche il codice più elegante e facile da leggere. La maggior parte dei linguaggi e dei **framework** ha inoltre già gli strumenti necessari quindi dovete solo scaricare le relative **immagini Docker** per essere rapidamente al lavoro. Detto questo, conviene sempre sovrastimare il tempo necessario quando si inizia a usare nuovi sistemi, per evitare ritardi in caso di problemi. Se presentate ai clienti o ai vostri superiori una **demo** di un progetto Open Source che l'ha implementata bene, l'importanza dell'**analisi statica** è abbastanza facile da comunicare per partire con questo tipo di procedura.

I benefici dell'analisi dinamica

Con l'espressione **analisi dinamica** si indicano tutti i test che si fanno solo sul software in esecuzione, come quelli di sistema, per le vulnerabilità, degli errori di memoria, delle prestazioni, della **concorrenza** e i **test di penetrazione** (**penetration test** o **pen test**). Anche se l'analisi statica fosse in grado di identificare tutti i problemi di sicurezza e i bug, potrebbe solo coprire il codice e le librerie. C'è sempre un modo per scrivere del cattivo codice che rispetta le regole e passa i test e ce ne sono vari per utilizzare male un programma corretto. L'analisi dinamica permette di identificare facilmente molti di questi problemi. Può aiutarvi a trovare bug come **Cookie-slack** (a causa del quale delle pagine che dovrebbero essere protette da un **login** sono accessibili anche se non c'è un **cookie** o è scaduto) e vulnerabilità come **Slowloris** (un tipo di attacco che, con pochissimo sforzo e traffico, può di fatto rendere inaccessibile la vostra applicazione), oltre a problemi di configurazione come supportare versioni non sicure dei protocolli **SSL** e **TLS**.
L'analisi dinamica, naturalmente, va oltre il collaudo **DAST** (**Dynamic Analysis Security Testing**). Alcune soluzioni che possono avere un impatto significativo sulla sicurezza sono pensate per migliorare la stabilità generale dell'applicazione, come il rilevamento degli errori di memoria e i test di carico e di sistema. L'uso di sistemi per la rilevazione di errori di memoria come **Valgrind** in **C/C++** è piuttosto comune, ma per linguaggi moderni come Python, **Rust** e **Go** questo tipo di strumento è sovente ignorato, perché si dà per scontato che il linguaggio possa gestire automaticamente questi problemi. Questo porta spesso a costrutti inefficienti al non accorgersi di **memory leak** che risultano evidenti solo quando ci sono forti carichi, ossia nel momento peggiore per rendersene conto. Se un negozio online **va in crash** nel giorno dell'inizio dei saldi, per esempio, non si limita a causare un mancato guadagno, ma è anche un rischio per la sicurezza. Se un malintenzionato scopre la possibilità di mandare in crash il vostro sistema causando una quantità di traffico trascurabile per un periodo di tempo prolungato, può far salire i vostri costi di **hosting** o bloccarvi il sito durante un evento importante.
La dura realtà è però che, anche se lentamente, l'analisi statica sta riuscendo a entrare nelle pipeline CI/CD, ma questo miglioramento nelle procedure di verifica non sembra estendersi al reame dell'analisi dinamica: la maggior parte delle aziende si limita all'occasionale pen test. Molte società tendono infatti a pensare "ci sono bersagli più interessanti di noi per gli hacker". Questo le porta a un falso senso di sicurezza e a fare nella migliore delle ipotesi due test di penetrazione all'anno, il che fa sì che problemi di sicurezza importanti possano rimanere trascurati per mesi.
I pen test in questione, inoltre, spesso consistono semplicemente nel fatto che un consulente esegua

un programma di scansione delle vulnerabilità come **Zed Attack Proxy** (**ZAP**), trovando una serie di problemi e non spingendosi oltre, visto che può già generare un corposo **report**.

A caccia di vulnerabilità

Cercare vulnerabilità ed **exploit** comuni in fase di produzione è importante ma, una volta che siete arrivati a questo punto, dovreste essere già ragionevolmente sicuri che non siano presenti. Come ottenere questo risultato? Anche se ci sono sempre più strumenti come **ZAP** che potete usare per fare un'analisi delle vulnerabilità della vostra **Web app**, non sono molti a sembrare pronti per la pipeline di integrazione e distribuzione continua così come sono. Si cominciano però a vedere delle proposte in questo spazio, tra cui una degna di nota è **StackHawk**, che è gratuita per gli sviluppatori di prodotti Open Source o che lavorano da soli e che è costruita a partire da una soluzione basata su **ZAP**. Uno degli aspetti che fanno spiccare questa opzione è che il fondatore di **ZAP**, **Simon Bennetts**, si è unito al suo team nel 2020. Potete creare una vostra soluzione personale basata su **ZAP** se non volete strumenti commerciali nella vostra pipeline, ma probabilmente a conti fatti vi costerà di più. Ci sono, naturalmente, altre soluzioni commerciali, come **Probely**, ma integrare queste piattaforme nella vostra pipeline CI/CD in genere richiede che esponiate l'ultima fase del vostro ciclo di sviluppo a degli **endpoint** pubblici, con le problematiche che ne conseguono. Con **StackHawk** e **ZAP** questo non è necessario, perché entrambi gli **scanner** si possono eseguire in un ambiente locale o in un **container**. Aggiungendo la ricerca delle vulnerabilità nella vostra pipeline di integrazione e distribuzione continua potrete cercare nella vostra Web app i problemi che gli hacker cercano di sfruttare, rendendo molto più difficile per gli attacchi automatizzati trovare punti deboli nelle vostre difese. Gli scanner rappresentano per l'analisi dinamica quello che i lint sono per quella statica: il minimo che potete fare per testare la vostra applicazione. Se, però, potreste anche pensare di pubblicare la vostra Web app con qualche problema di conformità minore (come un **docstring** mancante in una funzione), non è assolutamente accettabile che un'applicazione non passi un test di

L'interfaccia utente di **StackHawk** vi permette di risparmiare molto tempo nell'impostare le priorità

Vedere i cambiamenti nel tempo mentre lavorate sul codice può aiutare quando pianificate delle accelerazioni nel processo, perché vi aiuta a valutare quante **iterazioni** servono

vulnerabilità, perché significa che ci sono problemi che possono esporre i vostri server o i vostri clienti ad attività malevole o al furto di dati.

Creare software sicuro

Il primo passo per creare applicazioni sicure è capire la natura degli attacchi che potreste subire. Se avete dati personali o delle carte di credito, spesso sono i primi obiettivi di un malintenzionato. Se però avete un **blog** o un altro sito con contenuti statici o semistatici, è probabile che eventuali hacker cerchino di iniettare codice nelle vostre pagine per fare delle frodi ai danni degli utenti. Analizzare nel dettaglio le possibili minacce nella pianificazione delle funzioni del vostro progetto riduce il tempo per arrivare alla sua commercializzazione (**time to market**) integrando le protezioni necessarie nel suo **design** sin dall'inizio. Per fare un parallelo con i prodotti elettronici, aggiungere delle schermature alla fine del processo produttivo per proteggere il vostro dispositivo da interferenze elettromagnetiche è meno efficace e più costoso che progettarlo sin dall'inizio in modo da evitarle. Fare l'analisi delle potenziali minacce in fase di progettazione vi permette analogamente di risparmiare tempo e denaro. Integrare sia l'analisi statica sia quella dinamica nella vostra pipeline di integrazione e distribuzione continua dà a voi o ai vostri **product manager** i numeri e parametri per poter valutare costi e benefici di nuove funzioni o del refactoring. C'è anche un diretto beneficio per voi e il vostro team nel saper riconoscere il potenziale **debito tecnico** introdotto, per esempio, dalla vostra **Merge Request** (oppure **Pull Request** se vi trovate nell'**ecosistema Github**). Potrete infatti migliorare il **codebase** e la qualità, la sicurezza e l'affidabilità del progetto. La sicurezza delle Web app nasce da un modo diverso di pensare per i professionisti e da una cultura aziendale più lungimirante e attenta per le società: che vi troviate a lavorare per clienti o in ambito aziendale, molti miglioramenti si possono fare già con poco sforzo... si tratta solo di iniziare ad adottare le strategie migliori! LXP

THE X FILES

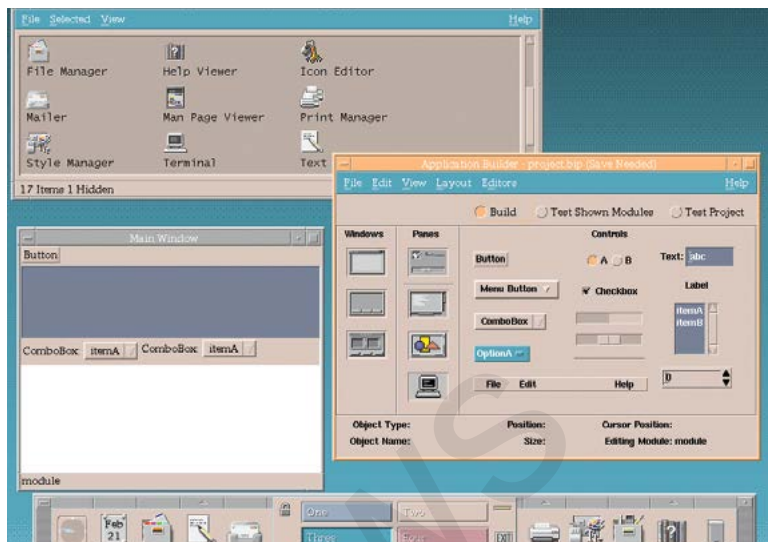
La storia e i segreti di un gestore grafico che da trent'anni si trova al cuore dei desktop con Linux

Ubuntu 21.04 segna il passaggio della distribuzione dal gestore grafico X.org a Wayland (sull'hardware compatibile), ma non è la prima volta che Canonical prova a usare Wayland di default. **Ubuntu 17.10** (**Artful Aardvark**) aveva questa configurazione, ma non è piaciuto agli utenti. Non solo tendeva a bloccare i laptop di **Lenovo** e ha dovuto avere una seconda **release**, ma ha deluso i fan del **desktop Unity** sostituendolo con **Gnome**. L'esperimento con Wayland non è proseguito con la successiva **versione LTS** (**Breezy Badger**) e non è presente neanche in quella attuale.
Ora la situazione si è stabilizzata e il prossimo Ubuntu LTS dovrebbe finalmente passare a Wayland. Ma vi serve davvero un nuovo **gestore grafico**? E cos'è esattamente **X**? L'argomento è complesso e ha una lunga storia alle spalle, ma con questo articolo potrete farvi un quadro più chiaro della situazione. Sicuramente, X è sul mercato da oltre trent'anni e non è ideale per gestire le applicazioni moderne. Anche Wayland però non è esattamente una novità. È stato concepito nel 2008 ed è stato utilizzato in vari dispositivi (non solo PC desktop con **Linux**) dall'inizio. È l'opzione predefinita in **Fedora** da anni e probabilmente funziona benissimo in altre distribuzioni: vale sicuramente la pena di scoprirlo!

X Window System è presente nei desktop con Linux dagli anni 90 del secolo scorso ed è nelle macchine UNIX (e Ultrix) sin dall'inizio del suo sviluppo nel 1984. È stato creato come parte del progetto Athena, uno sforzo congiunto del MIT e di DEC (Digital Equipment Corporation) con lo scopo di fornire agli studenti risorse informatiche.

Il nome **X** è ispirato da un suo predecessore, **W Window System**, creato per il sistema operativo sperimentale **V**. Il cambiamento principale intervenuto tra W e X è stato il passaggio dal **protocollo sincrono** a quello **asincrono**. Funzionalmente, X fornisce un'**interfaccia client/server** tra oggetti fisici (monitor, tastiera e mouse) e virtuali (il desktop).

Condividere le conoscenze

La storia della genesi di X inizia nelle università statunitensi. **Bob Scheifler** stava lavorando alla sua **tesi di master** al MIT e questa aveva come soggetto il **linguaggio di programmazione multithread Argus**. Scheifler voleva fare il **debugging** di più **thread** contemporaneamente e, dato che nel mondo **Unix** del 1984 non esisteva il concetto di **gestore delle finestre** o di **multiplexer del terminale**, doveva utilizzare diversi terminali fisici. Ai tempi non c'era neanche il concetto di Software Libero, ma c'era una cultura di condivisione tra le università e gli istituti di ricerca. Grazie a essa il codice di W (nato a **Stanford**) è finito nelle mani di Scheifler che ha deciso di farne il **porting** in Unix. Due settimane dopo aveva la prima versione di X in esecuzione. Questa edizione e le cinque successive sono state utilizzate internamente al MIT, ma anche in questo caso la cultura della condivisione lo ha presto fatto sfuggire ai confini dell'istituto. **X9** è stato pubblicato con la permissiva **licenza MIT** e diversi componenti erano solo distribuiti come **binari**.

CREDITI: Huihermit, Wikipedia/LGPL

Prima di **Gnome** e di **KDE**, c'era il proprietario **CDE**. Insieme al **toolkit Motif** che lo accompagnava, è diventato Open Source nel 2012

La versione **X10** ha visto la sponsorizzazione da parte di **IBM** del porting per i suoi nuovi modelli **RT PC** (**RISC Technology Personal Computer**) e per questa architettura era incluso quasi tutto il codice sorgente. Nel 1987 il protocollo di base era sostanzialmente standardizzato e noto con il nome di **X11**, utilizzato ancora oggi perché da allora non ha visto modifiche. Se siete utenti Linux da molti anni, potreste aver usato la versione **X11R6**, comparsa nel 1994. Per quella data c'era una nozione più formale del Software Libero e, trattandosi di un protocollo, erano disponibili diversi server X, liberi o proprietari. **XFree86** è nato come **fork** di uno di essi e si è imposto in Linux grazie al suo supporto per le schede grafiche per

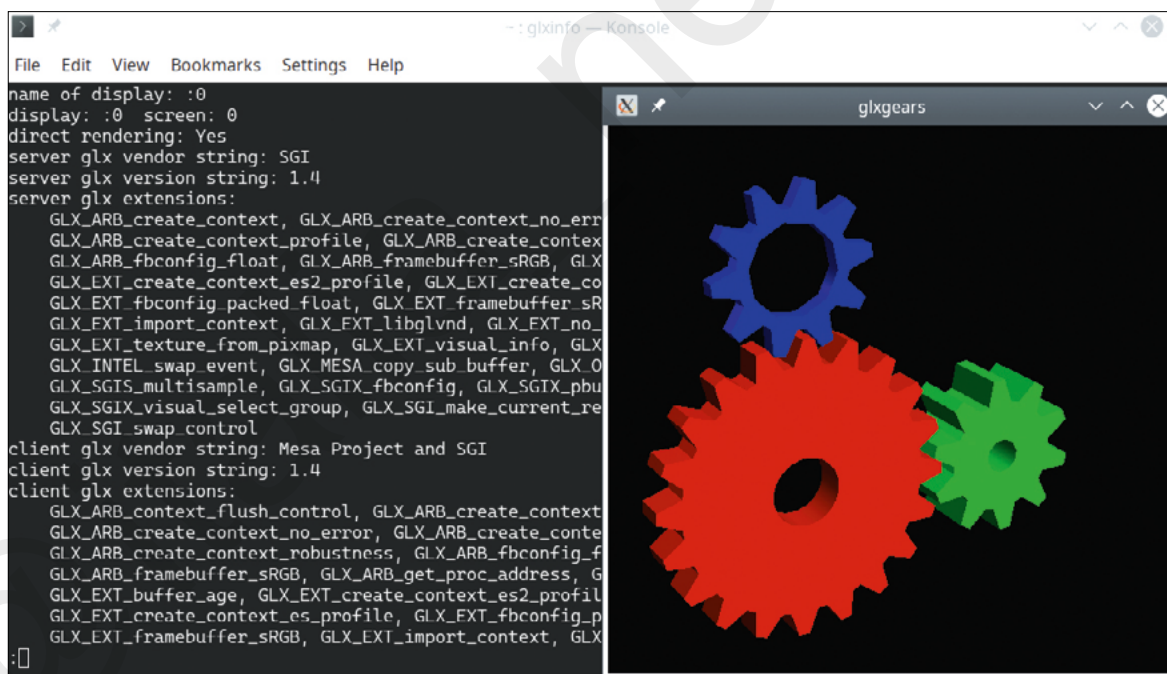
» TRASPARENZA DI RETE TRA PRESENTE E PASSATO

Uno degli elementi che rendono **X** unico è la **trasparenza di rete**, che va oltre lanciare i programmi in remoto con `ssh -X`, per quanto sia un comodo effetto collaterale. Un'applicazione X non sa se viene eseguita in locale o in remoto, una scelta che aveva senso all'epoca, dato che i computer desktop erano ancora relativamente giovani, ma che aggiunge un livello superfluo di complicazione ai carichi di lavoro odierni. Per questo, come ci ha spiegato **Daniel Stone** di **Collabora**, la trasparenza di rete non è presente in **Wayland** ed è stata una scelta deliberata. Malgrado questo, applicazioni come **XEDIT** o **Motif** funzionano in rete ancora meglio che con X, perché Wayland sfrutta il **core rendering** del server e altre tecniche. Dall'introduzione di **GTK2**, però, sono stati inviati al server solo **buffer pre-renderizzati**, quindi si trasmettono in rete buffer di **pixel raw**, il che è piuttosto inefficiente. Per questo soluzioni come la **tecnologia NX** (usata nel programma per desktop remoto **NoMachine**) sono molto utili, in quanto analizzano i dati in modo più intelligente e li comprimono di conseguenza. Il team di Wayland ha riconosciuto che non ha più senso far fare tutto il **rendering** al server a partire dalle **primitive** del **client**, perché è molto complesso. Invece di inviare buffer enormi, o far riconoscere al client se è in esecuzione in locale o in remoto e usare un percorso diverso, hanno abbandonato la trasparenza di rete. Neanche le versioni moderne di X (che usano l'interfaccia **DRI3** e la memoria condivisa) di fatto ce l'hanno, quindi non è un punto di confronto valido nei dibattiti sui due gestori grafici.

Daniel Stone di **Collabora** è un esperto di **X** e ha collaborato al progetto

PC. Un cambio di licenza nel 2004 lo ha però reso inadatto per molti progetti e incompatibile con la **GPL2** della **Free Software Foundation**. È nato così un nuovo fork più libero dal nome di **X.org**. Il componente server di X.org si chiama però **xfree86**. Il motivo per cui X11 è riuscito a rimanere utile per oltre tre decadi sono le sue estensioni, che la comunità ha iniziato a scrivere sin dalla sua nascita. **Ximage** e **Low Bandwidth X** (quest'ultimo mirato a migliorare le prestazioni sulle reti lente) sono state tra le prime. Se usate X attualmente potete vedere quali estensioni sono attive eseguendo **journalctl -b** e scorrendo le righe del suo **output**:

```
gdm-x-session[1149]: (II) Initializing extension Generic Event Extension
gdm-x-session[1149]: (II) Initializing extension SHAPE
gdm-x-session[1149]: (II) Initializing extension MIT-SHM
gdm-x-session[1149]: (II) Initializing extension XinputExtension
gdm-x-session[1149]: (II) Initializing extension DAMAGE
gdm-x-session[1149]: (II) Initializing extension RANDR
gdm-x-session[1149]: (II) Initializing extension GLX
```

Il nome dell'estensione **Damage** attira l'attenzione ma la sua funzione è abbastanza prosaica, dato che si limita a permettere alle applicazioni di tener traccia di quali parti delle loro finestre sono state aggiornate. Da solo, X traccia sullo schermo fisico nozioni astratte come finestre, riquadri, linee e curve. **Randr** permette di ridimensionare, ruotare e specchiare la **finestra di root** in modo da gestire i cambi di risoluzione o **display** aggiuntivi. Proprio per questo non c'è più bisogno di **sloggarsi** e **riloggarsi** per farlo. L'estensione **Composite** è stata introdotta nel 2004 e ha consentito ai **compositing window manager** (come **Compiz**) di creare effetti come trasparenze e le finestre che si distorcono (**wobbly window**), che sono diventate molto popolari in quel periodo. **Direct Rendering Infrastructure** (**DRI**) ha permesso alle applicazioni di accedere direttamente all'hardware grafico e di utilizzare l'accelerazione con i **driver Mesa OpenGL**, rendendo possibile l'utilizzo in Linux di alcuni giochi di punta, tra cui **Unreal Tournament 2004**. L'estensione **GLX** ha consentito alle applicazioni

Glxgears non è un **benchmark**, ma c'è qualcosa di maestoso nei suoi ingranaggi

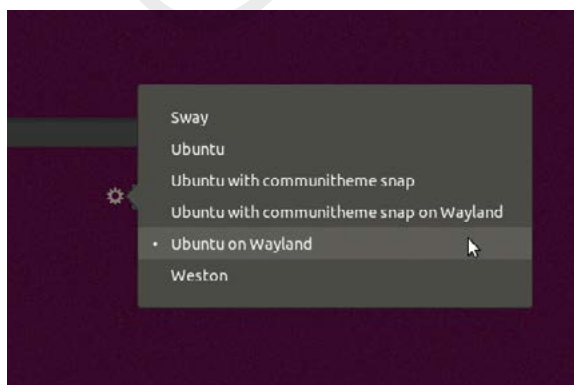
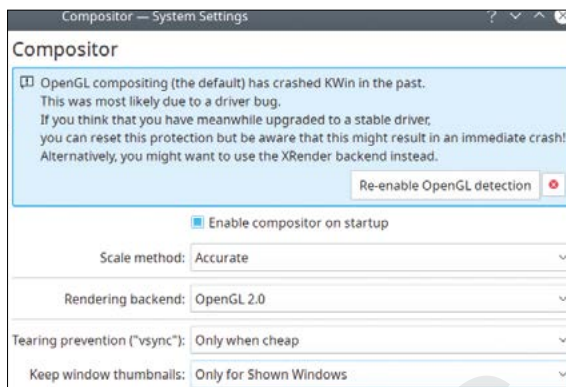
» UN APPROCCIO DIFFERENTE DAGLI ALTRI

Il posizionamento di **X** è sempre stato diverso da quello di altre **interfacce utente grafiche**. L'**Apple Lisa**, nel 1983, è stata la prima macchina ad averne una e un prodotto di nome **Windows** divenne molto popolare, per qualche motivo, dopo la sua introduzione nel 1985. Entrambe queste interfacce sono molto più vicine al sistema operativo di quanto X lo sia al **kernel** di **Linux**. Anche se le prime versioni di Windows si basavano sul **DOS**, il posizionamento di X e il fatto che è solo la parte di basso livello di ciò che normalmente si considera una **GUI** lo hanno sempre reso più complicato. C'è inoltre da considerare il groviglio di elementi **legacy** e di, a posteriori, cattive idee, che ha ereditato. Una di queste è la filosofia di utilizzare meccanismi al posto di indicazioni. Invece di dirvi come deve operare una GUI, accomoda (attraverso moltissimi meccanismi) ogni tipo di aspetto e comportamento e permette agli sviluppatori di applicazioni di scegliere quali usare. Oggi la situazione dei desktop per Linux può sembrare caotica, ma merita solidarietà il team di **freedesktop.org** che è riuscito a consentire a così tanti desktop diversi di interagire con infinite applicazioni.

OpenGL di disegnare direttamente in una finestra di X, in modo che gli utenti potessero godersi le mesmerizzanti rotazioni di tre ingranaggi colorati. Il programma **glxinfo** (dal pacchetto **mesa-utils**) fornisce informazioni sulle estensioni e i driver OpenGL. Un passo fondamentale per verificare che le capacità 3D della macchina funzionino correttamente consiste nel cercare la riga:

```
direct rendering: Yes
```

Non serve configurazione hardware

Nell'agosto del 2008 abbiamo intervistato **Keith Packard**, veterano sviluppatore di X, all'evento **Fosdem**. Allora X11 non era ancora in grado di avviarsi senza un file di configurazione **xorg.conf** che descrivesse **GPU** e monitor. Se cercate nei **post** dei **forum** dell'epoca, troverete domande su quali opzioni specificare per far funzionare bene i driver di **Nvidia** o su come ottenere l'accelerazione grafica 3D per migliorare le prestazioni di **Quake Glxgears**. Un errore di configurazione nei tempi del monitor in **xorg.conf** rischiava di rovinare la periferica ed è capitato anche qui in redazione che lo schermo iniziasse a produrre un rumore sempre più forte fino a quando non abbiamo staccato la spina.
Si può pensare che occorrenze come questa siano una cosa del passato e si spera che sia vero, ma la possibilità di rovinare gli schermi non è limitata a quelli analogici. X era in grado di rompere i monitor a tubo catodico, che ora sono abbastanza indistruttibili, ma Packard sottolineava che se programmate in modo sbagliato i **display a cristalli liquidi** interni potete farli alimentare con corrente continua, il che tende a far bollire il fluido al loro interno con la possibilità di compromettere in modo irreparabile il monitor. Oggi fortunatamente la maggior parte dei driver ottiene dagli schermi le informazioni necessarie a tutelarli ed è anche capace di aggirare dati sbagliati forniti da hardware difettoso. **Daniel Stone** di **Collabora**, che pure ha lavorato su X, ci ha raccontato alcuni dei suoi sviluppi e delle sue evoluzioni. Ci ha spiegato che molti dei progressi fatti dopo che X è arrivato a non aver bisogno della configurazione sono stati mirati a togliere più elementi possibile dal **server X**. **KMS** (**Kernel Modesetting**) è stato eliminato dai driver

In **Ubuntu** questa rotellina vi consente di scegliere tra una tecnologia di visualizzazione arcaica e il futuro

Il **compositor del desktop** di **KDE** permette dei bellissimi effetti e vi aiuta anche a risolvere eventuali problemi

dello **spazio utente** e la gestione dell'**input** è stata estratta in librerie separate. Anche se al momento queste misure miravano a migliorare X, stavano anche rendendo possibile sostituirlo. L'approccio è simile con Wayland: il **core** è molto piccolo e tutti gli sforzi recenti sono focalizzati su elementi come **PipeWire**, per lo **streaming video**, o miglioramenti come permettere di lavorare con **Flatpak**. Anche dopo un grande lavoro di **disaccoppiamento**, **refactoring** e rimozione di funzioni superflue, il **codebase** di X.org, dopo tre decadi di **plug-in** ed estensioni, è un incubo. Ben poche persone possono comprenderlo tutto e, citando lo sviluppatore di **Gnome Thomas Thurman**:

> **MONITOR A RISCHIO**
> "X era in grado di rompere gli schermi a tubo catodico, che ora sono abbastanza indistruttibili"

"Programmare X è come leggere uno di quei filosofi francesi che alla fine vi fanno chiedere se davvero sapete qualcosa di certo". Il server grafico era diventato analogo al suo sistema operativo. Un'altra preoccupazione relativa a X è la sua sicurezza.
Un'applicazione può manipolare la finestra di un'altra o prendere il suo input da tastiera. Uno dei programmatori che per lungo tempo hanno collaborato al kernel, **Matthew Garrett**, ha fatto un post nel suo **blog** sull'argomento nel 2016, evidenziando la presenza di un problema anche con le applicazioni **snap**, allora nuove, (vedi **https://bit.ly/3yhX5qx**). Ci sono strumenti che possono proteggere X in qualche misura, come **Security-Enhanced Linux** (**SELinux**), ma è difficile utilizzarli in modo efficace senza perdere funzionalità.
Con Wayland la memoria di un'applicazione non è accessibile alle altre e non c'è la necessità di gestire un milione di **comandi legacy** per il disegno, anzi non c'è nemmeno un'**API** per il disegno.
A Wayland interessano solo i suoi **buffer** pieni di **pixel**. Tutto diventa più lineare e a beneficiarne è anche il funzionamento della vostra macchina Linux. LXP

SALVA I DATI!

Con la giusta distribuzione potete diagnosticare e curare rapidamente diversi problemi di funzionamento di Linux

Linux ha un'ottima reputazione per quanto riguarda la stabilità, ma questo non significa che non possiate incontrare problemi. Che sia la vostra prima installazione o l'ultimo aggiornamento orchestrato del vostro cluster, a volte Linux smette di funzionare. Ci possono essere vari motivi, come guasti hardware, errore umano, sistemi operativi in conflitto, errori nel BIOS... e a volte sembra che l'unica via di uscita sia reinstallare tutto. Questa, però, non è mai una buona soluzione, in parte perché non vi permette di scoprire cosa sia andato storto, ma soprattutto perché dovete dedicarle molto tempo e impegno. Vale quindi la pena di conoscere le migliori distribuzioni mirate ad aiutarvi a capire cosa è successo nella vostra macchina **Linux** e a ripristinarne il funzionamento. Il **dual boot** con **Windows** è una causa comune di problemi quindi sarebbe meglio tenere Windows su un drive separato (se proprio vi serve averlo). Può capitare che un suo aggiornamento rimuova il **boot loader** di Linux dal vostro sistema. Questo problema si può risolvere facilmente (usando **SystemRescue** farlo è velocissimo) ma l'impatto che l'utente ha quando si verifica è che l'installazione Linux, con tutti i suoi dati, sia svanita del tutto. Sono anche stati segnalati casi in cui un **update** di Windows ha corrotto dei volumi Linux. La pratica migliore è quella di scollegare i drive Linux mentre il sistema operativo di **Redmond** si aggiorna ma, in ogni caso, è sempre meglio essere pronti a risolvere ogni problema con gli strumenti presentati in questo articolo.

Durante l'estate è stata messa in evidenza una nuova vulnerabilità di Windows. A causa di un problema con il suo spooler di stampa era infatti possibile scalare i privilegi e anche eseguire codice remoto. I dettagli della vulnerabilità (chiamata PrintNightmare) sono inoltre stati diffusi per errore (sul GitHub di Microsoft), aprendo le porte ai malintenzionati.
Il problema, per il quale sono state pubblicate delle **patch**, può sembrare irrilevante per gli utenti **Linux** ma in redazione abbiamo più macchine **dual boot** con Windows e gli aggiornamenti per risolvere il problema hanno fatto sparire il menu di avvio con **Linux**... e mezz'ora del nostro tempo a risolvere la situazione. Ritrovarsi con una macchina solo Windows dopo un aggiornamento di questo **OS** su un sistema dual boot non è un'esperienza rara, ma fortunatamente con il giusto arsenale questo tipo di situazione è facile da gestire.

Recuperare il boot loader

Il **boot loader** è uno dei punti più tipici in cui si verificano errori. Se la vostra macchina non lo trova (o non individua la **partizione EFI**), si comporta come se non ci fosse un sistema operativo. Il motivo può essere che dei dati si sono corrotti in un'interruzione dell'alimentazione o in un blocco del sistema, per esempio. Avviare una distribuzione mirata alla risoluzione dei problemi e al ripristino come quelle che trovate sul DVD virtuale è la soluzione migliore. Tecnicamente, una qualsiasi distribuzione (installata o che avete su una **chiavetta USB**) che vi consente di avviare in una circostanza simile può essere considerata "di ripristino", ma a volte servono strumenti specialistici. **SystemRescue** è un **toolkit** che vi permette di affrontare qualsiasi tipo di problema di sistema. Si avvia nel terminale, il che è comodo se la vostra scheda grafica ha dato **forfait**, ma si può lanciare un'interfaccia utente grafica scrivendo `startx`. Lo potete usare per ripristinare il menu di **GRUB** del vostro perduto dual boot, nel caso dovesse succedere. La chiave di questa operazione è il comando `grub-install`, che scrive GRUB nel **settore di boot** o crea una **voce EFI** appropriata.

SystemRescue offre molti strumenti di riparazione, ma qui bastavano un **terminale** e un comando poco noto

Non potete semplicemente puntare l'**installer** al vostro disco e sperare che tutto vada bene (come potete vedere in questa pagina, questo porta a un messaggio d'errore) perché GRUB deve scrivere dei dati di configurazione nel **filesystem**.
Il primo passo consiste quindi nel montare la partizione in cui GRUB era precedentemente installato. Potreste avere delle partizioni separate per l'avvio, **Home** e Windows, oppure averne una sola con la vostra distribuzione preferita. È importante in ogni caso identificarla correttamente, altrimenti potreste peggiorare la situazione. Qui viene usata la partizione fittizia **/dev/sdx1**:

```
# mount /dev/sdx1 /mnt
# grub-install /dev/sdx
```

Viene usato il prompt `#` per indicare che questi comandi sono eseguiti come **root**, perché SystemRescue usa solo l'account da amministratore. Se avete una distribuzione diversa potreste dover utilizzare `sudo`. A questo punto dovreste vedere il messaggio

```
Installation finished. No error reported.
```

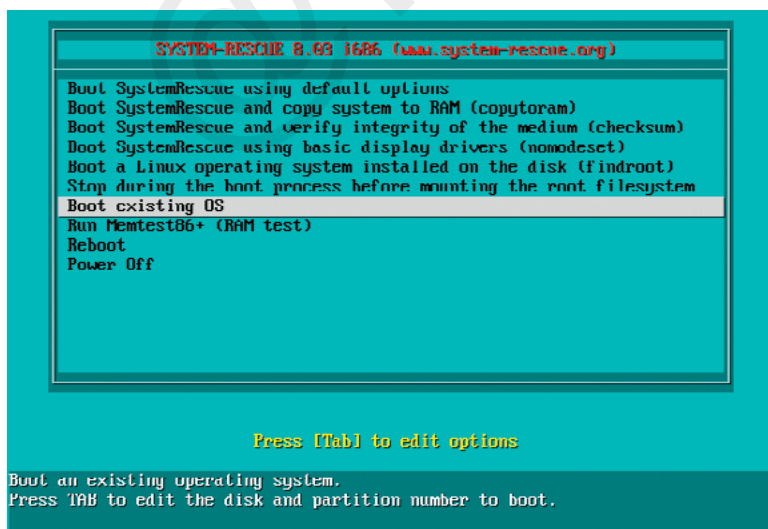
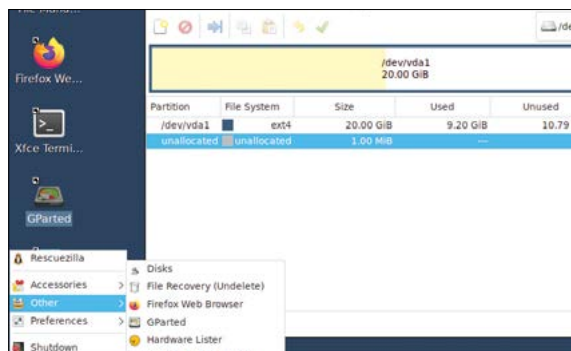
e riavviando dovreste risolvere il problema.
Il processo di boot potrebbe spingersi un po' più oltre, arrivando al menu di avvio, ma poi dovrebbe

» EVITARE I PROBLEMI INVECE DI RISOLVERLI

Si dice che prevenire è meglio che curare e l'adagio si applica anche al vostro computer. Molti problemi, infatti, si possono evitare con i giusti accorgimenti e avere **backup** aggiornati dei dati più importanti è un primo passo imprescindibile. Abbiamo dedicato uno speciale a questo tema nel **numero 204** di questa rivista, mostrando quanto è facile configurare **Déjà Dup** per automatizzare la procedura di backup. Se non volete impostare una **routine** strutturata, è comunque importante che copiate i file più importanti su qualche forma di servizio sul **cloud** (o, meglio ancora, nella vostra istanza di **Nextcloud**). Molti dei problemi descritti in questo articolo derivano da **Windows** e il nostro consiglio è di tenerlo su un disco separato se scegliete il **dual boot**. Molti esperti suggeriscono anche di scollegare i drive di Linux quando lo avviate. Staccare il **cavo SATA** ogni volta può sembrare eccessivo, ma può anche risparmiarvi notevoli grattacapi.
Se vi trovate a incappare in problemi con il **boot loader** di Windows, poi, non cercate di risolverli in Linux. **Ambiente ripristino Windows** ha il suo strumento **BootRec.exe** per reinstallare il boot loader e i dati di configurazione (**BCD**), ma non viene trattato in questa sede, in cui ci si focalizza su Linux. Potete trovare istruzioni in italiano per risolvere i problemi di avvio in Windows all'indirizzo **https://bit.ly/3z7EDkJ**.

portarvi al **prompt di ripristino** di GRUB (oppure a **systemd-boot**) quando provate ad avviare Linux. Potrebbe esserci anche il messaggio **error: unknown filesystem**. Il prompt di ripristino appare quando GRUB non riesce a trovare i suoi file sul disco. Può capitare dopo un aggiornamento di Windows perché quest'ultimo ha spostato delle partizioni, o per un errore di un'installazione di Linux. Sui sistemi **BIOS**, il primo stadio di GRUB viene installato nel **Master Boot Record** del disco, appena prima della **tabella delle partizioni**. Il file **stage1** dice al sistema dove trovare fisicamente lo **stage2** sul disco. Nei sistemi **UEFI**, viene usato il **caricatore EFI stub** per passare il controllo al boot loader su disco. In entrambi i casi il passaggio può essere destabilizzato dallo spostare file o partizioni sul disco. Una soluzione da provare al prompt di ripristino è dire manualmente a GRUB dove si trovano i suoi elementi. Come forse saprete, Linux etichetta i dispositivi come **nodi** del filesystem, per esempio **/dev/hda** (vecchi dischi), **/dev/sda** (drive più recenti) e **/dev/nvme0n1p1** (partizioni su **controller NVMe**). GRUB ha il proprio **sistema di indicizzazione dei drive**, a cui assegna nomi incrementali a partire da **(hd0)** (poi **(hd1)**, ecc.). Se un drive ha una vecchia tabella delle partizioni **MBR**, le partizioni vengono chiamate **(hd0,msdos1)** e così via. Le più moderne **GPT** sono indicate con **(hd0,gpt1)** ecc. In entrambi i casi, il primo

SALVATI DA GRUB

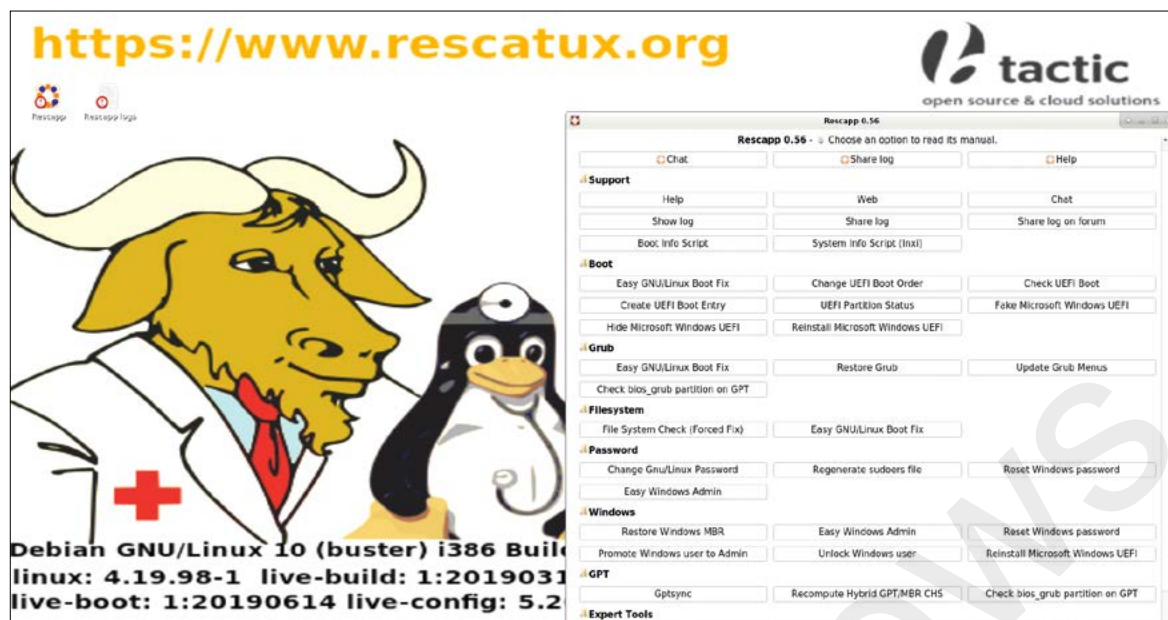
"Una soluzione da provare al prompt di ripristino è dire manualmente a GRUB dove si trovano i suoi elementi"

La schermata di **SystemRescue** che vi accoglie se avviate dalla sua **immagine ISO** originale

Oltre al programma principale **Rescuezilla**, nel menu trovate delle pratiche **utility** per il recupero e ripristino di dischi e file

dispositivo è il numero **0** ma la prima partizione è la **1**. Al prompt di ripristino, inserite **ls** per vedere la lista dei dispositivi e cercate per tentativi quale contiene la partizione di Linux. Quando trovate i numeri giusti, sostituiteli nei seguenti comandi:

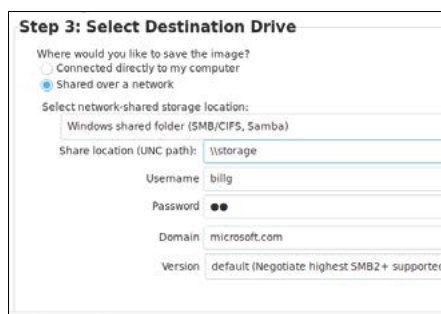
```
grub rescue> set prefix=(hd0,msdosx)/boot/grub
grub rescue> insmod normal
grub rescue> normal
```

L'ultimo dice a GRUB di avviarsi come farebbe normalmente e si spera che questa volta non siate rimandati al prompt di ripristino. Se dovesse succedere, ci sono altri metodi da provare.
Se il sistema si avvia normalmente, dovete rendere permanenti i cambiamenti eseguendo **sudo update-grub** dopo il **login**. Non è mai consigliabile cercare di modificare a mano il file di configurazione di GRUB (**/boot/grub/grub.cfg**). Il precedente comando ne genera uno nuovo. Se non dovesse funzionare potete reinstallare GRUB.

Resuscitare un hard disk

Le procedure di riparazione e ripristino non si limitano naturalmente alla risoluzione di problemi con i boot loader. Se tenete ai vostri dati, per esempio, dovete preoccuparvi del rischio che i drive possano guastarsi. Avere delle procedure di backup per i file più importanti è indispensabile (vedi box **Evitare i problemi invece di risolverli**), ma se prendete sul serio l'argomento vale la pena di considerare di fare il backup di intere partizioni. **Rescuezilla** si presenta come il coltellino svizzero per il ripristino dei sistemi ed è uno strumento utilissimo in diverse situazioni e facile da usare. Seguite i tre passaggi della guida nella prossima pagina per vedere come può essere semplice fare il backup delle partizioni. Potete ripristinarne una su un nuovo disco (per esempio, dopo un guasto hardware), con l'unica condizione che il supporto abbia abbastanza spazio libero.
Se si tratta di una partizione che avvia Linux, dovrete probabilmente reinstallare GRUB, come menzionato in precedenza, per ripristinarne il funzionamento. Potreste aver sentito parlare di filesystem moderni come **Btrfs** e **ZFS** e delle loro funzioni di riparazione automatica. Questa caratteristica può far pensare che usando queste tecnologie non dovrete più preoccuparvi della perdita di dati. Purtroppo, però, non è così. Prima di

Rescatux e il suo **front-end Rescapp** curano i problemi di computer con **Linux** da tempi immemori

tutto, nessun brano di codice avrà mai la capacità di proteggervi dalle vostre azioni e in secondo luogo tutto l'hardware prima o poi si guasta. Le funzioni di **self-healing** sono però un avanzamento significativo che, usando i **checksum**, mitiga il rischio che i dati si corrompano senza che ve ne accorgiate e vi permette di riconoscere l'hardware che si sta deteriorando prima che vengano fatti danni reali. Se incontrate un errore del disco, tenete presente che in genere non si tratta di incidenti isolati e che potrebbe essere il momento di considerare l'acquisto di un nuovo drive. Una cosa da non fare, se usate questi filesystem di nuova generazione, è cercare di ripararli voi stessi. Il comando **`btrfs check`** ha un'opzione **`–repair`**, che è pensata per l'utilizzo **offline**. Contiene numerose opzioni per ricostruire checksum e **alberi degli extent**, ma quasi sicuramente non vi aiuterà e, se usato male, può causare problemi. La stessa **man page** di Btrfs avvisa di non usare **`--repair`** a meno che non vi venga consigliato da uno sviluppatore o da un utente esperto. Se il filesystem si può sistemare, Btrfs lo fa in modo trasparente, altrimenti è meglio non peggiorare la situazione. Montate il dispositivo in sola lettura, scaricate i dati che potete prendere e trasferiteli su un nuovo supporto. Naturalmente queste pagine non esauriscono l'argomento della riparazione e del ripristino di Linux e ci sono molte distribuzioni valide. Per esempio, il venerando **Rescatux** vi permette di risolvere molti problemi comuni direttamente dalla sua interfaccia utente grafica. LXP

FARE IL BACKUP DELLE PARTIZIONI CON RESCUEZILLA

1 Avviate Rescuezilla

Se non avete il **DVD virtuale** di questa rivista a portata di mano, scaricate **Rescuezilla** dal sito **rescuezilla.com** e scrivetelo su una chiavetta USB con **Etcher** (**https://etcher.io**). Avviate quindi il vostro computer e attendete che compaia il semplice menu della distribuzione.

2 Fate il backup

Scegliete l'opzione di **backup** e selezionate il drive e la partizione di cui volete fare una copia di sicurezza. Potete fare il backup in locale o in remoto con **SSH**, **NFS** o **Samba**. Le immagini di Rescuezilla non vengono però compresse, quindi se la rete è lenta ci vuole molto tempo.

3 Ripristinate la partizione

Nel caso la partizione dovesse avere dei problemi e ogni altro tentativo di risoluzione risultasse inutile, potete ripristinarla, possibilmente su un nuovo disco. Scegliete l'opzione **Restore**, puntate Rescuezilla al file **.img**, ditegli dove metterlo e premete **Go** per dare inizio al processo.

@italia news

HAI PERSO UN NUMERO DI LINUX PRO?
NON PREOCCUPARTI PUOI ACQUISTARE GLI ARRETRATI!

Completa la tua collezione ordinando gli arretrati **a partire da 5,90€ cad.** su **www.sprea.it** oppure utilizzando il modulo qui sotto

SCEGLI L'ARRETRATO CHE VUOI ORDINARE

SE VUOI ORDINARE VIA POSTA O VIA FAX, COMPILA QUESTO COUPON

Ritaglia o fotocopia il coupon, invialo in busta chiusa a: **Sprea Spa** Via Torino, 51 20063 Cernusco s/n (MI), insieme a una copia della ricevuta di versamento o a un assegno. Oppure via fax al numero 02.56561221. Per ordinare in tempo reale i manuali collegati al nostro sito **www.linuxpro.it/arretrati**. Per ulteriori informazioni puoi scrivere a **store@sprea.it** oppure telefonare allo 02/87168197 dal Lunedì al Venerdì dalle 9.00 alle13.00 e dalle 14.00 alle 18.00.

INSERISCI I CODICI E MESI DI RIFERIMENTO delle pubblicazioni che desideri ricevere:

	€
	€
	€
	€
Totale Ordine	€

SCELGO IL SEGUENTE METODO DI SPEDIZIONE:
Indica con una ✗ la forma di spedizione desiderata

☐	Per **due o più riviste** spedizione tramite **Corriere Espresso** al costo aggiuntivo di	€ 4,90

TOTALE COMPLESSIVO	€

Data ______ Firma del titolare ______

Informativa e Consenso in materia di trattamento dei dati personali - (Codice Privacy d.lgs. 196/03) Sprea Spa Socio unico Sprea Holding Spa con sede in via Torino 51 – 20063 Cernusco s/ Naviglio (MI) è il Titolare del trattamento dei dati personali che vengono raccolti, trattati e conservati ex d.lgs. 196/03. Gli stessi potranno essere comunicati e/o trattati da Società esterne Incaricate. Ai sensi degli artt. 7 e ss. si potrà richiedere la modifica, la correzione e/o la cancellazione dei dati, ovvero l'esercizio di tutti i diritti previsti per Legge. La sottoscrizione del presente modulo deve intendersi quale presa visione, nel colophon della rivista, dell'Informativa completa ex art. 13 d.lgs. 196/03, nonché consenso espresso al trattamento ex art. 23 d.lgs. 196/03 in favore dell'Azienda.

NOME ______
COGNOME ______
VIA ______
N° ______ **C.A.P.** ______ **PROV.** ______
CITTÀ ______
TEL. ______
E-MAIL ______

SCELGO IL SEGUENTE METODO DI PAGAMENTO (Indica con una ✗ quello prescelto)

☐ Versamento su **CCP 99075871** intestato a **Sprea Spa arretrati Via Torino 51 20063 Cernusco Sul Naviglio MI** (Allegare ricevuta nella busta o al fax)

☐ Bonifico intestato a **Sprea Spa arretrati** sul conto **IBAN IT05 F076 0101 6000 0009 9075 871**

☐ **Carta di Credito** N. ______
(Per favore riportare il numero della Carta indicandone tutte le cifre)

Scad. ______ CVV ______
(Codice di tre cifre che appare sul retro della carta di credito)

______
Nome e Cognome del Titolare della carta (può essere diverso dall'abbonato)

☐

☐

☐

☐

☐

Il circuito integrato RP2040 va a ruba!

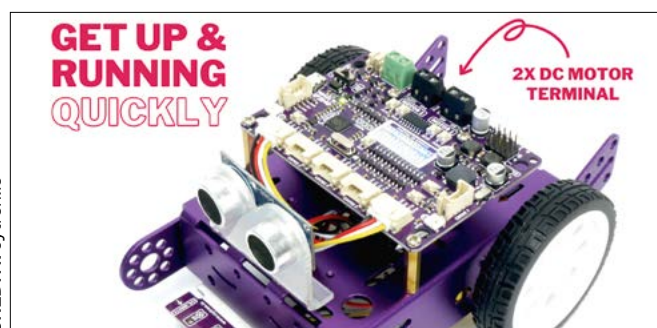
Ora che tutti possono comprare il cuore del Pico, i maker di tutto il mondo sono pronti a creare mille nuovi progetti

Al cuore del **Raspberry Pi Pico**, uscito all'inizio dell'anno, c'è il **circuito integrato RP2040**. Quest'ultimo era stato anche proposto in anteprima come componente singolo a una serie di aziende, il che ha permesso a **Pimoroni** e **Adafruit**, per esempio, di creare prodotti basati su di esso che sono usciti in contemporanea al Pico. Dopo il successo del lancio, però, la richiesta del circuito integrato da parte dei **maker** è stata così importante che il programma per fornirlo è stato anticipato e le prime unità sono state distribuite durante l'estate. Potete così comprare il **chip** per integrarlo nei vostri progetti e si trova nei negozi specializzati a poco più di un euro al pezzo. Sono già usciti molti prodotti che lo sfruttano, come per esempio l'**RP2040 Stamp** di **Arturo182** di **Solder Party** (**https://www.solder.party/**), una **scheda breakout** con un pollice di lato (2,54 cm) pronta da essere integrata nei progetti, il modulo **Wio RP2040** di **Seeed** (**https://www.seeedstudio.com/**) per portare il vostro RP2040 **online** e l'anello di rilevamento **CO2 RP2040** di **Guido Burger**, utile per misurare l'inquinamento ambientale, di cui trovate tutti i dettagli sul suo **account Twitter** (**https://bit.ly/3nskUdl**).

CREDITI: Raspberry Pi Foundation

Robot con l'RP2040

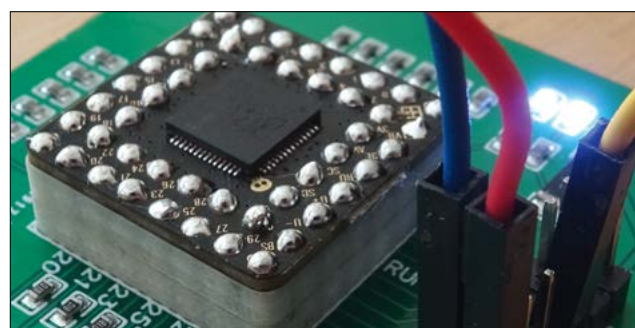
Una scheda controller dal cuore di Pi

Un altro progetto basato sul circuito integrato del **Pico** è il **Maker Pi RP2040** di **Cytron**. Offre una **scheda controller** con un **fattore di forma** pensato per i robot (88 x 64 mm), due terminali motore, quattro **porte servo**, **switch** di accensione e **reset** e molto altro per facilitare tutti i progetti con controllo remoto. Per i dettagli: **http://bit.ly/lxf278cytron**

CREDITI: cytron.io

Pronto per i vostri robot (progetto di esempio realizzato con la **Maker Pi RP2040**)

Accesso facile!

Una scheda breakout da Pimoroni

Con la **scheda breakout PGA2040** di **Pimoroni** potete avere facile accesso a tutti i **pin GPIO** dell'**RP2040**. Contiene solo gli elementi essenziali per il funzionamento dell'RP2040 e dovete collegarvi un **connettore USB** per programmarlo ma ha solo 21 mm di lato. Per informazioni: **https://shop.pimoroni.com/products/pga2040**

CREDITI: Chris Parrott/Pimoroni

Perfetto per i maker che non hanno il dono di dita sottilissime

@italia news

BACKUP

Backuppa e clona la tua Raspberry Pi

Che siano fatte clonando la scheda SD o copiando specifiche cartelle e partizioni, le copie di sicurezza fanno risparmiare tempo e sofferenza

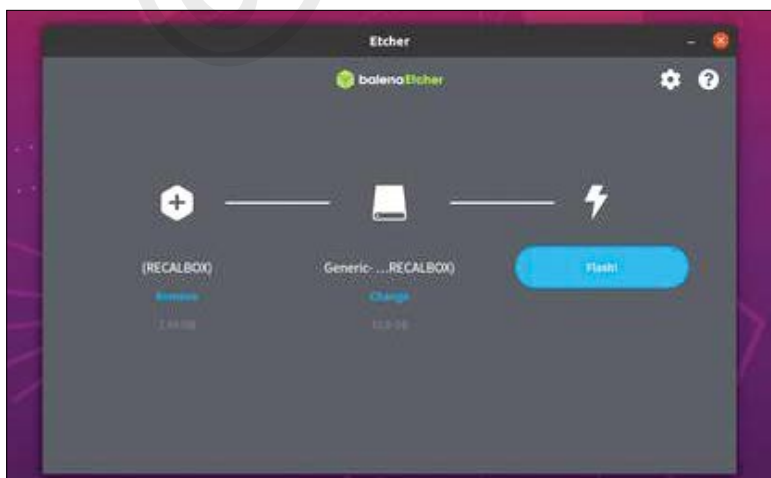
Anche se flessibile, economica e adorata, la **Raspberry Pi** non è immune ai problemi. Un suo punto debole è che salva su schede **SD** e **microSD** potenzialmente inaffidabili e che a volte perdono i dati dopo una chiusura o un riavvio improvvisi. Il miglior modo per evitare il problema è spegnere sempre correttamente la Pi, ma è anche utile avere un **backup** dei dati e ci sono varie opzioni per farlo, dal **clonare** la scheda microSD alle copie di sicurezza in remoto via **SSH**.

Le gioie della clonazione

Clonare la scheda SD vi permette di avere una libreria di immagini della Pi pronte all'uso sull'hard disk del vostro PC. Consente di risparmiare molto tempo, installando tutto il software e impostando tutte le connessioni una sola volta. Potete personalizzare **Raspberry Pi OS**, far funzionare tutto come vi piace e poi fare una copia della scheda SD da utilizzare più e più volte. Questo approccio, però, ha i suoi limiti. Prima di tutto, salvare più immagini occupa molto spazio di archiviazione. Dovete anche considerare il tempo che serve per creare l'immagine iniziale. Clonare semplicemente una nuova installazione di Raspberry Pi OS è inutile, perché la potete scaricare da **raspberrypi.org**. Deve invece includere sia il sistema operativo sia i programmi principali che volete usare e i vostri dati. Clonare la Pi può inoltre presentare delle sfide. Non potete, per esempio, clonare un'**immagine disco** da 16 GB e poi scriverla su una scheda SD da 8 GB, ma potreste avere dei problemi anche con una scheda SD da 16 GB, per le differenze negli standard di controllo qualità. È possibile anche trovarsi ad aver difficoltà ad avviare un'immagine disco clonata su un sistema diverso. Un backup creato con una **Raspberry Pi 4** in genere non funzionerà sulla **Pi Zero**. Se però fate attenzione alla compatibilità e alle dimensioni dei file, il resto della procedura è semplice e non presenta problemi. Potete clonare la scheda con un PC da scrivania o con una Raspberry Pi.

Invece di cimentarvi con il comando **dd** nel terminale, potete clonare le **schede SD** della **Pi** usando il programma **Etcher**

Gli strumenti che servono

Per creare il clone della scheda SD della vostra Pi con un desktop, vi serve solo lo strumento multipiattaforma per **flasharla**, **balenaEtcher** (**http://bit.ly/LXF287-etcher**). Scaricatelo, installatelo sul vostro PC e poi inserite la scheda microSD della Pi nel lettore del computer. Dovreste aver connessa una seconda scheda microSD pronta per la scrittura. Potete altrimenti usare una **chiavetta USB**. Non si può scrivere su un drive di sistema senza prima creare una partizione dedicata. Fate click su **Clone drive** e selezionate la sorgente (**Source**). Fate click su **Select** e poi su **Select target** per scegliere il drive di destinazione, che dovrebbe essere la seconda scheda microSD. Fate ancora click su **Select** per confermare e poi su **Flash** per cominciare. Attendete che il programma scriva e verifichi i dati. Potete anche clonare la scheda SD della Pi direttamente sul vostro hard disk, il che risulta utile se avete un solo **SBC** ma volete usarlo per molti progetti. Con la scheda SD inserita nel computer da scrivania, aprite il terminale e scrivete:

```
sudo fdisk –l
```

Identificate quindi la scheda SD e segnatevene il nome. Non è immediato ma potete riconoscerla abbastanza facilmente dalle sue dimensioni. Tipicamente il nome sarà **sda** o **sdb** seguito da un numero. Usate poi il comando `dd` per creare un clone. Potete dare al file di destinazione il nome che volete, ma deve avere l'estensione **.img**:

```
sudo dd if=/dev/sdb of=~/raspberry_pi_clone.img
```

Attendete il termine del processo. Quando siete pronti a scrivere l'immagine disco su una nuova scheda microSD, inseritela e scrivete:

```
sudo dd if=~/raspberry_pi_clone.img of=/dev/sdb
```

Potete anche usare **Raspberry Pi Imager**, lo strumento di per fare il **flashing** del sistema operativo disponibile sul sito **raspberrypi.org** (**http://bit.ly/LXF287-piimager**) per **Ubuntu**, **macOS** e **Windows**. Dopo aver installato il programma, fate click su **Choose OS**, poi scorrete fino a **Use Custom**, cercate il file .img clonato e scrivetelo su una nuova scheda SD.

Limitarsi alla cartella Home

Per una procedura più semplice e rapida, potete archiviare solo la **cartella Home** della Pi. Questo metodo vi permette di cercare e ripristinare cartelle e file in una nuova immagine della Raspberry Pi quando e se vi serve. Aprite un terminale e scrivete:

```
cd /home/
sudo tar czf pi_home.tar.gz pi
```

Con l'archivio creato, copiate **pi_home.tar.gz** su un altro dispositivo per il backup. La prossima volta che dovete creare una nuova immagine su una scheda microSD, semplicemente decomprimete la cartella **/home/** e copiate i file che vi servono sulla Pi.

Copie di sicurezza in rete

Se preferite usare la Raspberry Pi in remoto e avete il **protocollo SSH** abilitato, potete fare un backup in rete. Non è però il metodo ideale per l'intera scheda SD. Alcuni file in uso durante il funzionamento della Pi non vengono infatti clonati correttamente, lasciandovi con un'immagine disco che potrebbe non avviarsi. In rete è meglio salvare solo i dati principali, come la cartella Home. Per prima cosa avviate una sessione SSH e collegatevi alla Raspberry Pi. Inserite quindi il comando:

```
ssh pi@[INDIRIZZO_IP] "sudo dd if=/dev/mmcblk0 bs=4M | gzip –" | dd of=~/Desktop/[NOME_BACKUP].gz
```

Ricordatevi di sostituire `[INDIRIZZO_IP]` e `[NOME_BACKUP]` con quelli corretti nel vostro caso. Attendete quindi il completamento del backup. Naturalmente il ripristino non si può fare in rete, perché la Pi perderebbe la connessione durante la procedura. Per ripristinare l'immagine disco dovete inserire la scheda SD libera nel lettore del vostro computer e usare il comando `dd of`:

```
gzip -dc ~/Desktop/[BACKUP_ NAME].gz | sudo dd of=/dev/rdisk1 bs=4M conv=noerror,sync
```

I backup remoti via SSH fanno risparmiare tempo e non usurano la scheda SD della Pi. Sono un'opzione consigliata se avete più Raspberry Pi.

Raspberry Pi Imager offre varie funzioni nascoste, tra cui la possibilità di scrivere qualsiasi **file IMG** su una scheda **microSD**

Più spazio per i vostri archivi

Usare la clonazione per fare il backup dei file della Raspberry Pi occupa molto spazio su disco, soprattutto se eseguite la procedura su base regolare. Come impostazione predefinita, le immagini disco hanno esattamente le stesse dimensioni dei dischi, non dei dati utilizzati, quindi una scheda SD clonata da 8 GB occupa 8 GB di spazio sull'hard disk. Per fortuna i file si possono comprimere. Dovete solo inviare l'**output** di **dd** a **gzip** per creare un file compresso **.gzip**.

```
sudo dd bs=4M if=/dev/sdb | gzip > raspberry_pi_clone.img.gz
```

Per il ripristino usate quindi il comando:

```
gunzip --stdout raspberry_pi_clone.img.gz | sudo dd bs=4M of=/dev/sdb
```

Con questa tecnica il file clonato è più piccolo e occupa meno spazio sull'hard disk, dandovi uno stimolo in più per fare backup regolari del vostro prezioso lavoro... LXF

TIP

Usate le schede microSD più veloci, di classe A1. In questo modo la clonazione del sistema sarà più rapida e sicura, il che è utilissimo quando scrivete una nuova immagine disco. Cercate di scegliere marche note, per es. SanDisk, Kingston e Samsung.

» IL FILE DI CONFIGURAZIONE

Se la configurazione della vostra **Raspberry Pi** è particolare, magari a causa dell'hardware collegato, ma non vi servono **backup** regolari, invece di fare una copia di sicurezza dell'intera **scheda SD** o della cartella **/home** potete semplicemente fare un backup di **config.txt**. Questo file sostituisce il **BIOS** di sistema e contiene diversi parametri necessari per il corretto funzionamento della Raspberry Pi. Ogni suo **OS** ha un file **config.txt** che viene letto dalla **GPU** del sistema all'avvio del dispositivo. Con la Raspberry Pi accesa, potete modificarlo solo se avete **privilegi di root**. Lo trovate nella cartella **/boot** come **/boot/config.txt**. Con la **scheda microSD** inserita in altri sistemi, potete modificarlo con un **editor di testo**. Al suo interno si possono definire numerose opzioni. Ce ne sono di ogni tipo, dall'utilizzo della **RAM** alle impostazioni della videocamera, alle opzioni di visualizzazione, l'**overclocking**, il **GPIO** e molto altro. Se per esempio usate una **Raspberry Pi Zero** con un **display** ridotto, dovete modificare le opzioni predefinite di **config.txt**. Se lo salvate dopo i cambiamenti, in caso di reinstallazione potete semplicemente sostituire il file a quello **di default**.

ELETTRONICA

Tutti i segreti delle breadboard

Se vi state avvicinando ora al mondo della Raspberry Pi e dei single board computer, imparare a usare questo strumento è fondamentale

Uno dei punti di forza dei **computer a scheda singola** (o **SBC** da **Single Board Computer**) come la **Raspberry Pi** o **Arduino** è che hanno un connettore **GPIO** (**General Purpose Input/Output**) che permette di connetterli a dispositivi esterni, rendendo possibile la creazione di innumerevoli progetti. In genere, però, questi dispositivi esterni non si collegano direttamente ai **pin GPIO**, ma bisogna usare dei semplici circuiti come interfaccia. Se non avete mai lavorato con l'elettronica, potreste però avere difficoltà a individuare la corretta disposizione dei componenti, anche se avete uno schema del circuito, perché non è immediato tradurlo in pratica. Uno degli strumenti più importanti per farlo è la **breadboard**. Si chiama così (ossia letteralmente tagliere del pane) perché un tempo si usavano proprio dei taglieri di legno come base per i prototipi.

Perché usare una breadboard

Tradizionalmente i circuiti elettronici si creano saldando i componenti su un **circuito stampato**. Questo metodo non è però adatto alla fase di **prototipazione** e **testing**, per la quale spesso si usa una **basetta millefori**, ma anche in questo caso è necessario utilizzare un saldatore, il che significa che modificare il circuito e riutilizzare i componenti non è facile. Una breadboard offre una soluzione più pratica perché da un lato permette di creare anche circuiti complessi e dall'altra, non richiedendo saldature, facilita le modifiche. Si possono anche riutilizzare i componenti, anche se quelli usati per i prototipi non sono adatti al lavoro sul campo. In pratica, una breadboard è una basetta di plastica con una serie di fori disposti in righe e colonne e distanziati di 2,54 mm. Le dimensioni dei fori sono adatte all'inserimento dei **reofori** dei componenti più comuni e le molle sottostanti provvedono al loro fissaggio. Alcuni fori sono collegati tra loro in base a uno schema specifico (vedi immagine nella pagina accanto). Potete così connettere i componenti per formare un circuito semplicemente premendoli all'interno della scheda. Ci sono breadboard di diverse dimensioni, ma la più tipica è quella illustrata nel disegno, che ha 300 fori per i componenti. Le colonne sono definite con numeri da **1** a **30** e le file con lettere dalla **a** alla **j** quindi si può identificare univocamente qualsiasi foro. Il primo in alto a sinistra è per esempio **j1**. Potete usare queste definizioni per annotarvi una descrizione del circuito (per esempio, **resistore da 10K** da **j1** a **j6**) nel caso doveste ricostruirlo in seguito. Si possono inoltre trovare modelli di circuiti descritti in questo modo. Sopra i fori per i componenti ci sono altre due righe, una in alto e una in basso, di fori che non sono identificati individualmente. Le righe sono marcate con i segni **+** e **-** e spesso le aree sono delimitate da una riga blu e una rossa. Servono per l'alimentazione, tipicamente quella negativa, che si chiama **Ground** (**massa**) sulle Raspberry Pi e **GND** nelle schede Arduino, che si può anche trovare indicata come **0V** e a cui ci

CREDITI: George P. Macklin, CC BY-SA 2.0, www.flickr.com/photos/gmacklin/5079459828/

Questo esempio mostra che è possibile creare anche prototipi complessi con le **breadboard**, ma è meglio partire da qualcosa di semplice

TIP

Quando comprate un set di cavetti, tipicamente ne trovate di varie tinte. Usare colori diversi per scopi differenti rende più facile identificare i problemi se il circuito non funziona. Potete per esempio usare cavetti neri o blu per -ve e rossi per +ve.

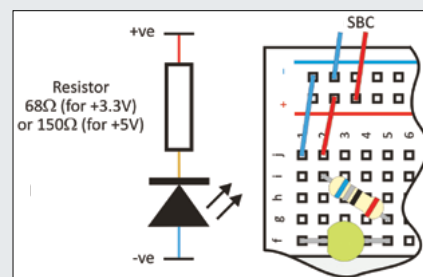
riferiremo come **-ve** nell'articolo. C'è anche l'**alimentazione positiva**, che può essere **+ 3,3 V** o **+ 5 V** in base all'SBC, qui chiamata **+ve**. Le linee dorate nel disegno rappresentano le connessioni interne e potete vedere che i fori di ogni riga dell'alimentazione sono connessi tra loro. Le linee +ve in alto e in basso non sono però collegate tra loro, né lo sono quelle -ve, quindi se volete usare sia quelle in alto sia quelle in basso dovete utilizzare dei cavetti di connessione. Ci sono molti fori per l'alimentazione perché sono ampiamente sfruttati nei circuiti. Tipicamente, se usate la breadboard per interfacciarvi con un SBC, dovrete connettere queste righe ai pin dell'alimentazione del computer a scheda singola con dei cavetti. Le linee dorate collegano anche, verticalmente e in gruppi di cinque, i fori per i componenti. Se dovete connettere più di cinque elementi, potete usare un cavetto per collegare un altro gruppo di cinque. Come si vede nel disegno, al centro della basetta c'è un'area senza fori e collegamenti, che è in genere una scanalatura. Il suo scopo è permettere l'utilizzo di circuiti integrati **DIP** (**Dual In-line Package**) che hanno un formato rettangolare con due file parallele di piedini separati da 7,62 mm o più, in base al numero di pin. Andrebbero fissati alla breadboard con le due file di piedini sui lati opposti della scanalatura. Le basette più grandi possono avere più di una scanalatura per consentire l'uso di più circuiti integrati DIP e spesso hanno interruzioni nella connessione delle righe orizzontali dell'alimentazione, che permettono di usare più fonti. A questo proposito, anziché usare una sola breadboard di grandi dimensioni, potete unirne varie più piccole e molti modelli sono predisposti per poterlo fare a incastro.

Una scorta di componenti

Il passo successivo è introdurre i componenti che volete utilizzare. In seguito potrebbero servirvene altri, ma il nostro consiglio è di partire da quelli più comuni se siete alla prime armi e volete iniziare a sperimentare per imparare e per creare dei semplici progetti. Quasi certamente utilizzerete dei **LED** di diversi colori e dei pulsanti, quindi questi sono i primi componenti che vi servono.
Per interfacciarsi con i LED servono dei **resistori**, ma il valore cambia in base al tipo e al colore del LED e alla tensione dell'alimentatore, quindi è utile averne un assortimento. Nel proseguire, vi serviranno anche vari **condensatori**, **diodi** e **transistor** oltre ad alcuni dei più comuni circuiti integrati. Avrete bisogno anche di cavetti per connettere la breadboard all'SBC. Per collegare i componenti tra loro servono **cavetti jumper maschio-maschio**, che si usano anche tra la breadboard e il GPIO delle schede Arduino.
Per collegarsi alla Raspberry Pi, però, servono cavetti **femmina-femmina**. Se siete alle prime armi vi conviene comprarvi un **kit elettronico di base**. Se ne trovano di diversi tipi a prezzi accessibili, ma verificate bene i componenti che contiene, perché possono variare sensibilmente. Uno molto popolare è il **Kit Elettronico con Breadboard** di **REXQualis** (**https://amzn.to/3mKCwkq**), che contiene circa 400 componenti tra cui resistori, LED, transistor e circuiti integrati confezionati

Le linee spesse dorate presenti in questo disegno non si vedono su una **breadboard** ma servono a rappresentare le connessioni interne tra i fori

» METTETE IN PRATICA LE VOSTRE CONOSCENZE

Un semplicissimo esercizio vi consente di mettere in pratica i principi esposti in questo articolo. Prima di tutto, costruite il circuito mostrato nella parte sinistra dell'illustrazione usando un **LED** rosso, giallo o verde, perché quelli blu o bianchi hanno bisogno di un **resistore** con un valore diverso, ma non fate la connessione rappresentata dalla riga rossa. Di solito i tratti negli schemi sono neri, ma qui gli è stato dato lo stesso colore dei cavetti corrispondenti nel disegno sulla destra, che è un segmento della **breadboard**. Dato che la connessione corrispondente alla riga dorata è fatta internamente dalla basetta, dovete solo fare i collegamenti rossi tra i fori e quelli rossi e blu tra la breadboard e l'**SBC**, che in questo caso agisce soltanto da alimentatore. Dopo che lo avete fatto, il LED non si accende, perché un'estremità del resistore non è collegata. Per farlo accendere, usate l'altro cavetto rosso per completare la connessione del resistore a **+ve**.
Per usare il computer a scheda singola per controllare il LED, scollegate l'estremità superiore del cavetto rosso che va dal resistore alla riga +ve della breadboard e collegatelo invece a un **pin** sul **GPIO** dell'SBC che si può usare come **output** digitale.
Scrivete infine un semplice codice per far passare a ogni secondo l'output digitale da **0** a **1** o viceversa e far lampeggiare il LED.

Questo semplice esercizio vi fa costruire il circuito sulla sinistra collegando i componenti mostrati sulla destra

singolarmente. Mancano, come spesso capita, alcuni elementi, tra cui i cavetti femmina-femmina, ma offre un ampio assortimento e, a meno di €15, un buon rapporto qualità prezzo. Nello scegliere il vostro kit di partenza fate attenzione alle recensioni e ai costi di spedizione. Un commento finale sui componenti riguarda i valori dei resistori, che sono indicati su di essi come fasce colorate. Alcuni kit includono una legenda ma, se non c'è, potete fare riferimento alla pagina di **Wikipedia** relativa (**https://it.wikipedia.org/wiki/Resistore**).

Componenti polarizzati

Molti componenti, come i resistori e vari tipi di condensatori, hanno solo due terminali e si possono connettere indifferentemente. Ce ne sono però altri che, pur avendo solo due reofori, devono essere connessi in un modo specifico perché sono **polarizzati**. Un esempio comune di questa categoria è un LED. Dato che probabilmente sarà uno dei primi componenti che utilizzerete, vale la pena di approfondire.
In particolare, ha un terminale positivo, l'**anodo**, e uno negativo, il **catodo**. Quando seguite lo schema di un circuito (detto anche **diagramma elettrico**) dovete essere in grado di riconoscere l'anodo e il catodo nel simbolo del LED e anche di identificare i due terminali nel componente reale. La figura in alto nella pagina accanto mostra il simbolo di un LED, con anodo e catodo identificati, e anche il componente fisico, sempre con i due terminali evidenziati. Noterete che il catodo nel componente fisico è identificato da un bordo piatto sulla superficie arrotondata del corpo del LED e anche dal fatto che risulta più corto dell'anodo. Fortunatamente, anche se collegate un LED con le polarità invertite non lo bruciate, a differenza di quanto accade con altri componenti, ma sicuramente non si accende.
A proposito di LED, vale la pena di sottolineare una criticità, anche se slegata dalla polarità. Se usate un **resistore limitatore di corrente** che ha una resistenza molto inferiore al valore raccomandato, distruggerete il LED. Altri due tipi di componenti polarizzati che incontrerete prima o poi sono i diodi e i **capacitori elettrolitici**, di capacità più elevata degli altri modelli a parità di dimensioni. Li trovate nella stessa illustrazione nella pagina accanto, dove potete vedere che il simbolo di un diodo è simile a quello di un LED (che è in realtà un tipo di diodo) ma senza le due freccine, quindi una volta che sapete individuare anodo e catodo nei LED potete farlo anche nei diodi. Nel componente fisico, il catodo è identificato da una fascetta sul corpo del diodo. Nell'illustrazione sono mostrati due simboli diversi per i condensatori polarizzati perché ci sono standard differenti. Fate particolare attenzione quando collegate i condensatori elettrolitici: se li montate con la polarità sbagliata possono, infatti, arrivare a esplodere.

CREDITI: pixabay.com

Per costruire i vostri circuiti elettronici vi serve una serie di componenti di uso comune. Comprare un kit elettronico di base è una soluzione pratica ed economica

» SFRUTTARE L'ATTREZZATURA PER IL TESTING

Ci sono due strumenti utilissimi per testare il vostro circuito su **breadboard** e trovarne i problemi.
Il primo è la **sonda logica**. Assomiglia a una penna con due cavetti da un lato e una sottile punta metallica dall'altro. Dopo aver collegato i due cavi con i loro morsetti a **-ve** e **+ve**, toccando una parte del circuito con la punta farete accendere il **LED** rosso o verde per mostrare lo **stato logico 1** (+ve) o **0** (-ve) rispettivamente. L'altro strumento è il **multimetro** (o **tester**) che, come il nome suggerisce, può misurare diverse caratteristiche elettriche. Prima di tutto, può fare **test di continuità**: emette un segnale acustico quando c'è continuità elettrica nel tratto di circuito in esame, provando se due punti sono collegati. Il tester misura inoltre la **resistenza** di ciò che collegate ai suoi terminali. Se, quindi, li connettete al **reoforo** di un **resistore**, vi conferma se è del valore corretto. Se però provate a misurare la resistenza di un componente nel suo circuito, è possibile che il valore sia falsato da altri componenti. Il multimetro può infine misurare la **corrente** anche se, per rilevarla in un punto del circuito, dovete interromperlo in quel punto in modo da poter collegare il tester al circuito **in serie**.

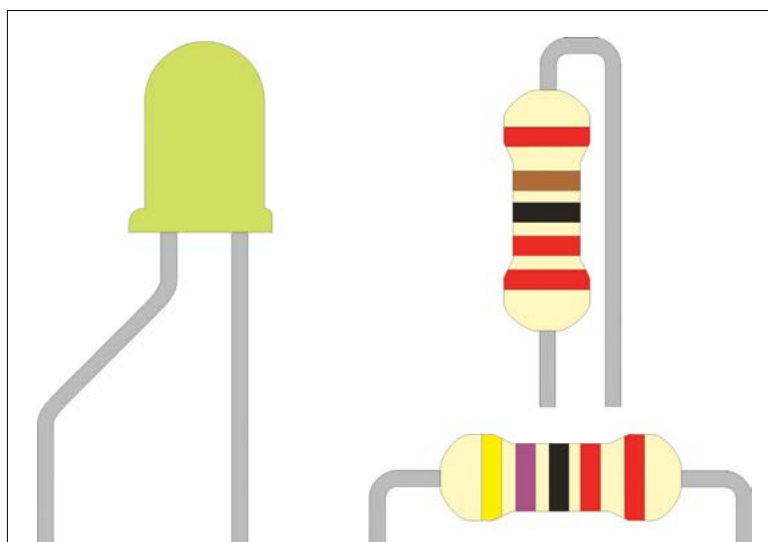
Un multimetro vi permette di misurare **continuità**, **resistenza**, **tensione** e **corrente** ed è uno strumento utile e poco costoso

Come costruire un circuito

Passiamo ora a vedere come costruire un circuito, ma in questa sede non entreremo nel merito della scelta di quali componenti inserirvi e di come collegarli tra loro. La progettazione di un circuito, infatti, è un argomento completamente separato e molto articolato che non può trovare spazio in questo articolo. All'inizio, d'altro canto, realizzerete circuiti che qualcun altro ha progettato ma di cui non ha necessariamente pubblicato il **layout** per la breadboard. Con il tempo, poi, potrete progettare i vostri circuiti, il che dà grandi soddisfazioni e vale sicuramente l'impegno necessario. Partite però interfacciando componenti comuni come LED e interruttori per poi procedere gradualmente a creare progetti più articolati. Conoscendo le connessioni interne della breadboard, le dimensioni dei componenti che dovete utilizzare e la spaziatura tra i loro terminali, è relativamente facile capire la distribuzione dei componenti per l'implementazione di un dato circuito. Malgrado la semplicità del processo, ci sono alcuni aspetti pratici da tenere presenti. I LED hanno **terminali radiali**, il che significa che entrambi protrudono dallo stesso lato del corpo del componente. Potete quindi montarli direttamente in un paio di fori adiacenti sulla breadboard. Se però i fori sono troppo distanti tra loro i reofori non riescono a raggiungerli quindi dovete piegare uno di essi o entrambi a un angolo di 45 gradi. I resistori, invece, hanno **terminali assiali**, il che significa che protrudono dai lati opposti del corpo del componente. Non c'è quindi la possibilità di montarli sulla breadboard senza piegare almeno uno dei due reofori. Un metodo per risolvere è piegare solo uno dei due terminali di 180 gradi in modo che risulti parallelo all'altro, che è solo a una breve distanza da questo, permettendo di montare il resistore in due fori adiacenti della breadboard, tenendone il corpo in posizione verticale. L'altro metodo è piegare entrambi i reofori di un angolo a 90 gradi in modo che puntino nella stessa direzione, consentendovi di fissare il resistore in due fori più lontani tra loro, mantenendo in questo caso il corpo del componente orizzontale. Piegare i terminali vicino al corpo di un componente rischia di danneggiarlo se non lo fate in modo corretto. È meglio usare delle pinze a becco sottile e piegare il reoforo verso l'esterno rispetto al corpo. Quando smontate il circuito sulla breadboard, non conviene raddrizzare i terminali dei vari componenti. Alcuni potrebbero infatti servirvi con le stesse angolazioni e, se piegate e raddrizzate un terminale troppe volte, alla fine si rompe. A volte non è possibile costruire un circuito semplicemente inserendo i componenti nella breadboard perché i loro terminali hanno una lunghezza limitata. Le connessioni interne della breadboard non sono però l'unico modo per collegare i componenti e potete usare anche dei cavetti. Servono, naturalmente, anche per connettere il circuito sulla breadboard ad attrezzatura esterna, nella maggior parte dei casi una Raspberry Pi, una scheda Arduino o un single board computer analogo. Prima di testare il vostro circuito, dopo averlo realizzato, controllate bene che sia stato implementato correttamente, perché potete bruciare dei componenti se non lo è. Anche se appare a posto a un'analisi visiva, se non sembra funzionare quando lo provate staccatelo subito dall'alimentazione prima di procedere a cercare di capire cosa c'è di sbagliato. A questo punto, oltre ad analizzarlo visivamente, conviene usare della semplice attrezzatura per il **testing**. In particolare un **multimetro** semplifica sensibilmente il processo di individuazione dei problemi. Fate riferimento al box **Sfruttare l'attrezzatura per il testing** per ulteriori dettagli. Quando inizierete a creare circuiti più complessi, potrete anche usare degli strumenti software per aiutarvi a progettare il **layout** della breadboard, ma non sono necessari per fare i vostri primi esperimenti e progetti. LXP

Alcuni componenti con due terminali vanno collegati in un modo specifico. Qui viene mostrata l'identificazione dei diversi terminali sia sui componenti fisici sia negli schemi per i **LED** (sinistra), i **diodi** (centro) e i **condensatori polarizzati** (destra)

TIP

Non è consigliabile, ma si possono costruire anche circuiti complessi sulle breadbord. Pare che sia persino stato fatto il prototipo di una scheda grafica in questo modo, anche se il risultato è stato un groviglio di cavi.

Dovrete probabilmente piegare i terminali di alcuni componenti. I **LED** (sinistra) si possono montare su due fori se sono vicini, ma se sono più distanti dovete piegare almeno uno dei **reofori**. Per i resistori va sempre piegato almeno un terminale e ci sono due modi per farlo

Recensioni

Tutte le novità in campo software e hardware testate e valutate ogni mese dai nostri laboratori

Se vuoi segnalarci qualche novità scrivi a redazione@linuxpro.it

Una breve legenda

Ogni test di questa sezione è accompagnato da un giudizio che riassume con quattro indici numerici le principali qualità dell'applicazione o del prodotto hardware messo alla prova. I laboratori di Linux Pro assegnano un voto da 1 a 10 alle seguenti categorie:

Caratteristiche: fornisce tutte le funzioni di cui abbiamo bisogno? È innovativo?
Prestazioni: esegue in maniera efficiente le sue funzioni? È veloce e affidabile?
Facilità d'uso: dispone di un'interfaccia grafica chiara e facilmente fruibile? La documentazione che lo accompagna è sufficientemente completa ed esaustiva?
Qualità/prezzo: ha un prezzo competitivo? Vale i soldi richiesti per il suo acquisto?

Il nostro giudizio viene poi riassunto da un voto finale, espresso anche graficamente. Ecco la legenda dei voti:

10 Nulla da eccepire. Un prodotto praticamente perfetto.
8-9 Un buon prodotto. I pochi difetti presenti non sono gravi.
6-7 Compie il suo lavoro ma necessita di ulteriori sviluppi.
4-5 Deve migliorare prima di raggiungere un voto sufficiente.
1-3 Un completo disastro. Gli sviluppatori devono tornare alla fase di progettazione.

Ricordiamo infine che i software citati nelle sezioni Confronto e Da non perdere sono spesso presenti nel DVD sotto la voce "Rivista" sotto forma di codice sorgente o binario.

QUESTO MESE...

Desperados 3

Intel Core i5 11600K

OSGeoLive 14.0

Desperados 3

La pianificazione tattica è il vostro pane quotidiano e sognate una carriera da pistoleri nel Far West? Non perdetevi questo titolo...

SPECIFICHE

Minime
OS: Ubuntu 16.04+ (64 bit)
CPU: 3,5 GHz Dual Core (64 bit)
Memoria: 8 GB
GPU: Nvidia GTX 560, Radeon HD 5850, 2 GB di VRAM
HDD: 21 GB

Consigliate
OS: Ubuntu 20.04 (64 bit)
CPU: 3,5 GHz Quad Core (64 bit)
GPU: Nvidia GTX 760, AMD Radeon HD 7870, 3 GB di VRAM

Desperados 3 ha un'ottima selezione di strumenti utilissimi, dalla tagliola che intrappola con i suoi denti chiunque ci passi sopra ai dardi magici che vi consentono di uccidere due nemici alla volta, ma a permetterci di portare a casa la pelle nei nostri test sono spesso stati gli umili tasti **F5** ed **F8** per salvare e ricaricare. Ogni sfida del gioco è un vero **puzzle** in cui spesso dovrete procedere per tentativi. Il cuore di Desperados 3 è il lavoro di squadra di un gruppo di specialisti altamente sincronizzati che usano precisione e tempismo per superare situazioni impossibili, ma la via del successo è costellata di catastrofi. Basta sbagliare di un secondo o un centimetro per trasformare un capolavoro di collaborazione in un bagno di sangue ma, con qualche ricaricamento, ce la farete. Magari il vostro tentativo di uccidere un nemico facendogli cadere sopra un muro è stato sventato da una guardia che vi ha riconosciuti all'ultimo minuto. Riprovate mandando un complice travestito a distrarre i nemici. Un'altra guardia lo pesca comunque ma, al tentativo dopo, mettete una trappola per fermarla. Il suo cadavere, d'altro canto, attira l'attenzione, sventando i vostri piani. La volta successiva, lasciate qualcuno nascosto tra i cespugli per prendere e occultare la salma... Per approssimazioni successive, arriverete al piano perfetto. Alcune missioni si svolgono in zone civili in cui riuscite a esplorare la mappa senza farvi notare.

Potete così origliare le conversazioni per ottenere indizi, identificare i vostri obiettivi e trovare modi per ucciderli. Gli indizi però sono piuttosto semplici (del genere "c'è un cartello in bilico sopra la testa di quel tipo") e non vi portano lontano: passate la maggior parte delle missioni avanzando di soppiatto e procedendo a suon di uccisioni. Non che far cadere un cartello in testa a qualcuno non abbia i suoi meriti: sfruttare la situazione e l'ambiente per liberarvi dei vostri avversari è più divertente che limitarsi a spararli. C'è in genere anche un fattore di rischio in questo approccio, il che lo rende ancora più stimolante.

I morti non parlano, per quanto possiate impegnarvi

Basta un pizzico di magia

A circa un terzo del gioco, incontrate **Isabelle** e scoprite il mondo della magia. Possedere animali, portare le guardie a uccidersi tra di loro, collegare le persone in modo che subiscano la stessa sorte... ogni missione vi spinge a manipolare i nemici per portarli in posizioni in cui potete eliminarli o procedere senza essere individuati, ma di solito è fatto indirettamente. Un personaggio che può controllare quasi tutti gli esseri umani e gli animali è quindi una ghiotta opportunità. Non ha un'arma a distanza e compare in poche missioni, altrimenti non usereste nessun altro. La sua magia, infatti, si innesta perfettamente in un gioco basato interamente sul pensiero laterale, ma incantesimi e **cowboy** sono un mix inusuale. Sfortunatamente non viene sfruttato nella trama, che è un semplice racconto di vendetta unito alla storia delle origini del gruppo. Nessuno dei suoi membri è riuscito a farci avere una risposta emotiva, ma le loro prestazioni all'opera sono ottime. La gioia di trovare sempre nuovi metodi per sfruttare i loro mille talenti e vedere la perfetta esecuzione dei piani

rimane per tutto il gioco. La modalità **Showdown** consente di fare attacchi sincronizzati, permettendovi di mettere in pausa il gioco per dare a ogni personaggio un ordine, per poi attivarli singolarmente o tutti insieme. Potete così creare complesse sequenze mantenendone il pieno controllo. Il problema è solo aspettare il momento giusto, ma si risolve facilmente con la funzione di avanzamento rapido che si può usare in qualsiasi momento e probabilmente sfrutterete spesso.
La combinazione di armi da fuoco, numerose casse di munizioni e la possibilità di mettere in pausa l'azione vi permette di smettere di agire di nascosto e attaccare apertamente, senza preoccuparvi di causare allarme e trasformando il gioco in uno sparatutto tattico. Anche se preferite essere silenziosi e creativi, è sempre utile avere l'opzione di uscire da una brutta situazione in questo modo.

Limiti e confronti con altri titoli

Tutto è cresciuto in Desperados 3: ci sono più mappe, più nemici, più esplosioni... oltre a molti più ostacoli e altrettanti metodi per superarli. Il gioco, però, si affida un po' troppo alle sue dimensioni e ai suoi numeri per creare le sfide. Nella parte iniziale, per esempio, c'è una missione in cui dovete difendere una piccola fattoria da un attacco ma, invece di focalizzarsi su quella singola posizione, l'azione si espande su una vasta area in cui ancora una volta dovete uccidere e riuscire a superare senza farvi scoprire un piccolo esercito. Anche se ci sono delle belle aree, nessuna delle mappe è spettacolare. C'è una gran quantità di fango, sabbia ed edifici sgangherati e, anche se il viaggio del vostro gruppo lo porta in diverse regioni, molte mappe sono prive di una propria identità visiva, soprattutto in confronto a quelle di **Shadow Tactics**, in cui luci e grafica erano determinanti per creare un'atmosfera. Ci sono, però, degli ottimi treni e a un certo punto potete incendiare una grossa imbarcazione in mezzo a una palude. La campagna può offrirvi circa 25-30 ore di avanzati enigmi tattici, ma ogni missione ha anche una serie di sfide nascoste. Quelle di velocità vi impongono obiettivi da raggiungere in 15 minuti per missioni che richiedono ore per il completamento. C'è anche una serie di sfide separate che riutilizzano le mappe

L'abilità nel poker non è la sola virtù di **Kate O'Hara**

introducendo nuovi obiettivi e variazioni. Tutto questo rende il titolo molto longevo, anche perché avrete sempre voglia di riprovare l'emozione di risolvere scenari omicidi o di attraversare una mappa in un quarto d'ora senza farvi pescare. Desperados 3 non vi assegna un punteggio ma siete sempre stimolati a darvi nuovi obiettivi ed esplorare strade che non avete mai percorso. Avrete sempre in mente nuovi piani che avreste potuto applicare a una missione precedente. Il gioco è più grande e vistoso di Shadow Tactics, anche se un po' meno elegante, ma il tutto è coerente con il tema di pistoleri e fuorilegge, che fa pensare più alla dinamite che alle sottigliezze. Anche fallire in sfide elaboratissime in città di frontiera, fattorie e paludi è stato divertente e arrivare a risolverle ci ha fatti sentire i pistoleri più strategici del West. LXP

VERDETTO

SVILUPPATORE: Mimimi Games
WEB: https://desperadosgame.com/
PREZZO: 40,99 €

GIOCABILITÀ	**9/10**	**LONGEVITÀ**	**9/10**
GRAFICA	**8/10**	**QUALITÀ/PREZZO**	**9/10**

Desperados 3 trasforma il Far West in uno brillante scenario per dimostrazioni di abilità tattica e strategica sempre ricche di stimoli.

» Il voto di Linux Pro 9/10

Intel Core i5 11600K

Un buon rapporto qualità/prezzo e tanta potenza per i videogame!

SPECIFICHE

Socket: FCLGA1200
Arch.: 64 bit
Core: 6
Thread: 12
Processo: 14 nm
Clock: 3,9 GHz, (boost: core singolo 4,9 GHz, tutti i core 4,6 GHz)
Cache: 12 MB di Smart Cache
Memoria: DDR4-3200, due canali, no ECC, 128 MB max
PCIe: v4.0, 20 linee
TDP: 125 W
iGPU: Intel UHD Graphics 750, OpenGL 4.5, OpenCL 3.0

Uno dei migliori modelli per chi ama i videogame dell'undicesima generazione di **CPU Intel**, il **Core i5 11600K** ha una **velocità di clock** di 4,6 GHz con tutti i **core** e di 4,9 GHz con uno singolo, superando in entrambi i casi il **Comet Lake i5**. La famiglia di processori **Rocket Lake**, però, non è solamente una versione del Comet Lake con velocità di clock superiori: è una nuova architettura.

Passi avanti su più fronti

Le **istruzioni per ciclo** (**IPC**) sono aumentate di circa il 19 per cento rispetto ai precedenti processori centrali di Intel per desktop e sono stati introdotti dei miglioramenti per il supporto allo **standard PCIe 4.0**, uno dei punti deboli dei modelli della casa. Il vantaggio di questa nuova CPU sono le sue migliori prestazioni, mentre lo svantaggio è che si sono persi i benefici dei modelli con **processo costruttivo a 10 nm**, con il risultato di avere un **chip** più grosso, più incline al surriscaldamento e più oneroso in termini di assorbimento energetico. Con i videogiochi, l'i5 11600K offre generalmente le stesse prestazioni dell'**AMD Ryzen 5600X** e regge quasi il confronto con l'**i9 11900K**, che costa circa il doppio. Le **performance** analoghe a quelle del 5600X si sono confermate nelle altre prove con **benchmark**. È leggermente in svantaggio con i test **x264** and **Cinebench** ma supera decisamente il concorrente per quanto riguarda la memoria. L'unico fattore in cui c'è una differenza significativa tra il Core i5 11600K e il Ryzen 5 5600X è l'assorbimento energetico. L'architettura **Zen 3** con processo a **7 nm** offre un'ottima efficienza con un picco di assorbimento di soli **76 W**. Il Rocket Lake i5 arriva invece a **130 W** e molto di più se lo spingete con carichi di lavoro **multithread**. Questo è il principale compromesso causato dall'adozione del processo costruttivo a **14 nm**. La differenza di circa 40 € tra i prezzi dei due processori non rappresenta però un vantaggio enorme per Intel. Il 5600X è infatti una CPU più efficiente con una gestione migliore dello standard PCIe 4.0. D'altro canto il i5 11600K è il processore Intel che più tiene testa a questo modello, anche se i compagni di scuderia **Core i5 11600KF**, più economico e senza **GPU Xe**, e **Core i5 11400**, anch'esso con sei core e 12 thread, sono dei buoni prodotti. In ultima istanza, il principale svantaggio dell'i5 11600K rispetto al 5600X è la minor efficienza energetica, che è raramente la prima preoccupazione per chi vuole una macchina da gioco. LXP

Le **istruzioni per ciclo** (**IPC**) sono aumentate di circa il 19% dai modelli precedenti e la **frequenza di clock** può arrivare a 4,9 GHz

La competizione tra **Intel** e **AMD** fa migliorare la qualità delle **CPU**

VERDETTO

PRODUTTORE: Intel
WEB: www.intel.com
PREZZO: 260 €

CARATTERISTICHE	9/10	FACILITÀ D'USO	9/10
PRESTAZIONI	8/10	QUALITÀ/PREZZO	9/10

Uno dei miglior chip Core i5 proposti da molto tempo, perfetto per il mercato dei PC per i videogiochi e in grado di trarre il massimo da ogni scheda grafica.

» Il voto di Linux Pro 9/10

Fedora 34

La community Open Source ha avuto qualche screzio con le distribuzioni legate a Red Hat recentemente, ma questa è davvero difficile da criticare

IN BREVE

Una delle più amate distribuzioni RPM continua a giustificare la sua fama di miglior distro con l'ambiente desktop Gnome. Molti dei suoi sviluppatori sono pagati da Red Hat, che usa il progetto per affinare i componenti che in futuro entreranno nei suoi prodotti commerciali

SPECIFICHE

Minime
CPU: 2 GHz
Memoria: 2 GB
Spazio su hard disk: 20 GB
Build: 64 bit x86, aarch64

Fedora 34 è stata la prima grande distribuzione ad adottare **Gnome 40**, il cui salto di numero rispetto alla precedente **v3.38** è in parte mirato a semplificare la nomenclatura dell'**ambiente desktop**, ma porta anche delle novità. Quando entrate in Fedora 34 venite portati a una schermata **Activities Overview** aggiornata. Gli sviluppatori sono inoltre passati da un **layout** verticale a quello orizzontale, quindi la **Dash** è in basso e gli spazi di lavoro sono organizzati in una striscia orizzontale. Altri aspetti dell'ambiente desktop sono stati modificati per renderlo più intuitivo ed efficiente. La Dash, per esempio, che elenca sia i programmi preferiti che gli altri, ora li separa visivamente. Anche le miniature delle finestre sono state migliorate e ora aggiungono le icone dei programmi per aiutarvi a identificarle. Gnome 40 è basato sul nuovo **toolkit grafico GTK 4.0** che, a parte le modifiche esteriori, ha portato diverse migliorie al funzionamento interno. È per esempio stato ampliato il supporto delle **gesture**, permettendo agli **swipe** con tre dita di cambiare spazio di lavoro e caricare la panoramica delle attività. In generale, l'esperienza utente è molto migliorata.

Con **Fedora 34** gli sviluppatori hanno introdotto diverse migliorie e cambiato il processo di creazione dell'immagine per l'installazione, riducendone le dimensioni

Cosa c'è sotto il cofano

Come sempre, il desktop è basato sul **display server Wayland**, anche se gli utenti di hardware **Nvidia** vengono passati automaticamente al più vecchio e lento **X.org**. Gli sviluppatori stanno però lavorando per permettere anche a loro di usare Wayland. Un importante cambiamento nel **back-end** di Fedora 34 è stato il passaggio dai **server audio PulseAudio** e **Jack** a **PipeWire**. Dato che le **API** di quest'ultimo sono compatibili con quelle di PulseAudio, la maggior parte dei programmi funziona senza alcuna modifica. Gli sviluppatori di Fedora hanno inoltre lavorato con **Google** e **Mozilla** per assicurarsi la piena compatibilità con i **browser Chromium** e **Firefox**. Un'altra novità è che il **filesystem Btrfs** di Fedora 34 ora fa la **compressione trasparente di default** con **zstd**. Oltre a far risparmiare tra il 20 e il 40 per cento di spazio di archiviazione, questa modifica permette anche di aumentare la longevità di **SSD** e altri **dispositivi flash**. La **release** principale **Fedora Workstation** è progettata come una distribuzione per desktop estremamente solida, adatta in particolare agli utenti avanzati che la usano come piattaforma per compiti come la programmazione o l'elaborazione di file multimediali. Apprezzeranno modifiche come l'abilitazione di **systemd-oomd** per aiutarli a evitare **errori di memoria OOM**. Oltre all'edizione Gnome, l'ecosistema di Fedora ne produce altre per rispondere alle esigenze di ogni tipo di pubblico, con diversi ambienti desktop e applicazioni. Ci sono anche varianti per i differenti casi d'uso. C'è una versione della distribuzione per i server, una per gestire i contenitori (**CoreOS**) e un'altra per dispositivi dell'**Internet delle cose** (**IoT**) che offre un miglior supporto per un'ampia gamma di **device** con **architettura ARM**. LXP

VERDETTO

PRODUTTORE: The Fedora Project
WEB: www.fedporaproject.org
LICENZE: varie

CARATTERISTICHE	8/10	**FACILITÀ D'USO**	9/10
PRESTAZIONI	8/10	**DOCUMENTAZIONE**	9/10

Un'altra ottima release di Fedora stable che raggiunge il perfetto equilibrio tra forma e funzione.

» Il voto di Linux Pro **8/10**

OSGeoLive 14.0

Ora che le vacanze sono decisamente finite, è tempo di scoprire una distribuzione che può portare tutto il mondo a casa vostra

IN BREVE

Distribuzione basata su Lubuntu che offre alcuni dei migliori strumenti geospaziali Open Source oltre ai relativi dataset. Gli sviluppatori li hanno preconfigurati in modo che siano subito utilizzabili. OSGeoLive è supportato da numerose organizzazioni.

OSGeoLive è una distribuzione basata su **Lubuntu** che raccoglie una libreria di strumenti e applicazioni **geospaziali** ed è finalizzata solo a questo tipo di utilizzo. Con il termine geospaziale ci si riferisce a tutte le tecnologie usate per acquisire, manipolare e conservare informazioni geografiche. Strumenti di questo tipo sono utilizzati da chi fa ricerca o analisi di dati, dai professionisti della **cartografia** e dagli amatori delle mappe e non mancano nel mondo Open Source. OSGeoLive ne include vari, preconfigurati per diversi casi d'uso e resi accessibili anche per i neofiti. Il progetto è sostenuto dalla fondazione non-profit **OSGeo Foundation**, che supporta anche lo sviluppo di vari strumenti geospaziali Open Source, e viene proposto anche per uso didattico e applicazioni pratiche.

Il desktop include un elenco dei nomi utente e delle password predefiniti per le varie applicazioni che trovate preconfigurate nell'applicazione

SPECIFICHE

Minime
CPU: 1 GHz
Memoria: 1 GB
HDD: 20 GB
Build: 64 bit, ISO e VM

Esploratori seduti alla scrivania

OSGeoLive è distribuito come un'**immagine ISO** che si può copiare su un **DVD** o un supporto **USB**. In questo secondo caso potete anche salvare tutti i cambiamenti che fate nell'**ambiente Live**. Il progetto propone anche un **disco virtuale vmdk** che si può collegare a **macchine virtuali VMware** e **VirtualBox**. La distribuzione accetta solo progetti Open Source solidi, ma include circa 50 strumenti geospaziali oltre a varie mappe mondiali gratuite e **dataset**. Trovate i programmi specialistici nella **metacategoria Geospatial**, suddivisa in gruppi come **Desktop GIS**, **Navigation and Maps**, **Web Services** e altri. Ogni categoria contiene alcuni strumenti e, anche se non sono adatti all'utente casuale, il fatto che siano preconfigurati li rende rapidamente accessibili a chiunque. Ci sono classici programmi **Geographic Information System** (**GIS**) da scrivania per vedere, elaborare e analizzare dati geospaziali e **client** basati su **browser**. La categoria Navigation and Maps include strumenti più noti come il mappamondo virtuale **Marble**, **OpenStreetMaps** e i suoi **editor JOSM** e **iD**. Accanto agli strumenti geospaziali ci sono diverse fonti di dati gratuiti per le mappe. Ce ne è per esempio una che fornisce ai cartografi mappe di pubblico dominio che li aiutano a creare le proprie mappe regionali e nazionali, e un'altra che include i dettagli delle temperature e precipitazioni registrate nel passato e di quelle previste per il futuro. Oltre a dei miglioramenti alla documentazione e alle traduzioni, l'ultima **release** ha aggiunto anche dei nuovi strumenti. Per la gestione dei pacchetti, la distribuzione si affida al **package manager Muon**, che usa i **repository** predefiniti di **Ubuntu** oltre al **PPA** di OSGeoLive Per l'installazione viene utilizzato l'**installer** standard di Lubuntu, che è **Calamares**. OSGeoLive è una di quelle distribuzioni che sono più della somma delle loro parti. Non è solo una raccolta dei migliori strumenti geospaziali Open Source, ma anche un metodo che permette agli utenti di accedervi in modo rapido ed efficiente. LXP

VERDETTO

PRODUTTORE: OSGeo Foundation
WEB: https://live.osgeo.org/en/index.html
LICENZE: LGPL

CARATTERISTICHE	8/10	FACILITÀ D'USO	8/10
PRESTAZIONI	8/10	DOCUMENTAZIONE	8/10

La distribuzione ideale per chi si interessa a strumenti geospaziali, professionalmente o per hobby.

» Il voto di Linux Pro 8/10

Aorus Gen4 7000s

L'ultimo SSD di Gigabyte sa farsi notare per le sue ottime prestazioni

SPECIFICHE

Capacità: 2 TB (1 TB)
Fatt. forma: M.2 2280
Interfacce: PCIe 4.0 x4, NVMe 1.4
Logica: Phison PS5018-E18
DRAM: DDR4
Memoria: Micron 96L TLC
Lett. seq.: 7.000 MBps 2 TB, 1 TB
Scritt. seq.: 6.850 MBps (5.500 1 TB)
Lett. casuale: 650.000 IOPS (350.000 1 TB)
Scritt. casuale: 700.000 IOPS 2 TB e 1 TB
Sicurezza: AES 256 bit
Endurance: 1.400 TB (700 TB 1 TB)
Garanzia: Cinque anni

Con il suo **SSD Aorus Gen4 7000s**, **Gigabyte** ha incluso un ottimo **dissipatore di calore**, che mira a ridurre le temperature del 20% e si rivela molto utile in un dispositivo che può arrivare a un assorbimento di **8,5 watt** sotto carico pesante. Questo modello è pensato per competere con i migliori **SSD** e può arrivare a **7 GBps**. Ci sono due capacità: **1 TB** e **2 TB**, entrambe con una velocità di **7.000 MBps** per la **lettura sequenziale**. Mentre il primo modello però in scrittura arriva a **5.500 MBps**, il secondo raggiunge i **6.850 MBps**. Entrambi hanno una garanzia di cinque anni e una buona **endurance**. Il dissipatore incluso misura **11,5 x 23,5 x 76 mm** e l'SSD si trova tra due spessi **pad termici**. Il dispositivo si basa sul **controller SSD PCIe 4.0 x 4** di **Phison** di seconda generazione, il **PS5018-E18**. Sfrutta la memoria **DRAM** e ha un'architettura a **triplo core**, accompagnata dalla tecnologia **CoXProcessor 2.0** dell'azienda (altri due **core R5**) per ottenere tempi di risposta veloci e coerenti. I core principali della **CPU** sono **Arm Cortex R5** con 1 GHz di **clock**, superando i **733 MHz** del modello precedente, il **PS5016-E16**.

Le prestazioni nei nostri test

Nelle nostre prove l'Aorus si è posizionato primo per velocità in lettura e secondo per rapidità nella copia, superato solo dal **980 PRO** da 2 TB di **Samsung**. Ha riportato punteggi eccellenti anche in scrittura, raggiungendo circa **5.850 MBps**. In termini di risposta nei test di **lettura e scrittura casuale** si è rivelato veloce quanto il **Rocket 4 Plus** di **Sabrent**, ma non ha raggiunto Samsung, **WD** o **Adata**. I test di **scrittura sequenziale** mostrano però che l'Aorus è uno dei modelli più veloci sul mercato. Nelle nostre prove ha scritto **464 GB** di dati a una velocità d**i 6,9 GBps** prima di scendere a **1,8 GBps** per altri **1,18 TB** e di arrivare a una media di **1.150 MBps** per il resto del test. Dal punto di vista dell'assorbimento energetico risulta più efficiente dal Rocket 4 Plus, ma meno di altri modelli Phison per assorbimento medio. Arriva a picchi di **8,5 watt** mentre a riposo l'assorbimento è ben regolato a **800mW**, il secondo valore più basso nei nostri test, battuto solo dal Samsung 980 Pro. Nelle nostre prove **in idle** l'SSD aveva una temperatura di 40° senza flusso d'aria mentre dopo aver copiato metà della sua capacità ha raggiunto la temperatura superficiale di 78° senza **thermal throttling**. Per chi cerca un SSD con dissipatore che può tener testa ai migliori modelli, l'Aorus Gen4 7000s di Gigabyte è un prodotto valido. LXP

Il **controller** di seconda generazione di **Phison** fa un ottimo lavoro

VERDETTO

PRODUTTORE: Gigabyte
WEB: www.aorus.com
PREZZO: 400 € (210 € il modello da 1 TB)

CARATTERISTICHE	9/10	FACILITÀ D'USO	9/10
PRESTAZIONI	9/10	QUALITÀ/PREZZO	8/10

Ha tutte le caratteristiche di un SSD NVMe di alto livello, con buoni valori di velocità nei test e anche un'endurance di tutto rispetto.

» Il voto di Linux Pro 9/10

Il confronto

Ogni mese mettiamo a confronto prodotti e programmi per farvi scegliere al meglio!

Modellatori 3D

Non tutti hanno il tempo per imparare a usare del software 3D complesso. Ecco cinque semplici soluzioni per creare bella grafica tridimensionale

MODALITÀ DEL TEST

Abbiamo utilizzato estensivamente ognuna delle applicazioni testate, seguendo i tutorial relativi e consultando la loro documentazione ufficiale, sia per imparare a usarle sia per valutare il supporto della comunità e le guide disponibili. Le applicazioni per il **3D**, infatti, non servono a molto se non potete apprendere facilmente come usarle. Abbiamo valutato quale sarebbe stato l'impatto di ogni progetto per un utente privo di esperienza e anche le sue prestazioni, l'affidabilità e il flusso di lavoro. Nessuno dei prodotti testati richiede un computer potentissimo, ma per tutti è necessaria l'**accelerazione hardware 3D**. Per il **rendering** è inoltre utile avere una **CPU** veloce. **Bforartists** è basato su **Blender** e, se il vostro obiettivo è usarlo per creare il prossimo **Toy Story**, vi serve una macchina potente. Se invece volete solo imparare le basi e fare lavori di complessità standard, non avrete problemi.

Le applicazioni per la **modellazione 3D** non devono necessariamente essere complesse da usare. Qui ne vengono valutate 5 di semplice utilizzo, adatte per chi vuole avvicinarsi al mondo della grafica tridimensionale senza investire moltissimo tempo per diventare un esperto. **Blender** non è stato incluso perché non mira alla semplicità, ma è stato esaminato **Bforartists**, un suo **fork** con un'interfaccia più lineare. Viene anche valutato **Art of Illusion**, che copre funzioni simili a quelle di Blender ma non ha una curva di apprendimento altrettanto ripida e non offre un labirinto di funzioni, menu ed **editor**. **Asset Forge** ha un funzionamento originale: l'utente unisce gli elementi predeterminati forniti per creare degli oggetti come veicoli e costruzioni. **Dust3D** è progettato per creare elementi organici come animali e persone ed animarli, ma ha anche la capacità di realizzare oggetti inanimati. **Goxel**, infine, adotta un approccio diverso dagli altri strumenti, utilizzando i **voxel**, che sono l'equivalente 3D dei **pixel**, per consentirvi di creare figure 3D fatte di blocchi in stile **Minecraft**.

@italia_news

Flessibilità e usi potenziali

Cosa potete creare col programma?

Le applicazioni esaminate possono essere di uso generico o specifico. **Goxel**, per esempio, svolge un singolo compito molto bene, ma se non volete creare modelli **blocchettosi** basati sui **voxel** è inutile. Analogamente, **Dust3D** si focalizza su un duplice tema, ossia la creazione di elementi organici (come animali e persone) e la loro animazione basata su uno scheletro che rende i movimenti realistici. Ha anche però il potenziale per creare forme inanimate come una bicicletta. **Asset Forge** è pensato per realizzare semplici modelli, come veicoli, che si possono poi usare in altro software e anch'esso non è molto flessibile. Questi strumenti danno il meglio di sé quando usati per progettare oggetti da importare in altri modellatori più avanzati o in un **motore di gioco**. **Art of Illusion** è invece un'applicazione completa, utile a chi vuole creare modelli 3D senza investire tempo a imparare a usare **Blender**. **Bforartists** è il programma più versatile, perché ha tutte le funzioni di Blender per modellazione, **sculpting**, animazione e **rendering**. È ideale per realizzare giochi e grafica in 3D e persino per la produzione cinematografica. Offre anche funzioni di base per l'elaborazione video e il disegno in **2D**. Il tempo per imparare a usarlo, però, è superiore a quello necessario per altri programmi. Sia **Art of Illusion** sia Bforartists possono svolgere anche le funzioni delle applicazioni più specialistiche. Potete per esempio usarli per creare oggetti blocchettosi con i voxel anche se, nel caso dobbiate utilizzare molto questo stile, è più facile sfruttare Goxel come editor e poi importare i modelli realizzati con esso in un programma più articolato.

Dato che **Bforartists** è un **fork** di **Blender,** può sostanzialmente affrontare qualsiasi compito per la grafica in **3D,** ma richiede più impegno degli altri

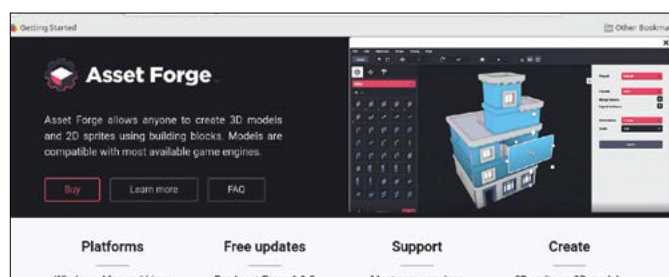
VERDETTO

BFORARTISTS	**9/10**	**DUST3D**	**4/10**
GOXEL	**4/10**	**ASSET FORGE**	**4/10**
ART OF ILLUSION	**7/10**		

È quasi impossibile battere un derivato di Blender per quanto riguarda la flessibilità nel mondo della modellazione 3D.

Licenze e installazione

Che ostacoli bisogna superare prima di iniziare a lavorare?

Molti dei programmi esaminati (tutti tranne **Asset Forge**) sono Open Source e gratuiti. Anche **Asset Forge** ha comunque un costo contenuto. Dopo il pagamento, vi viene inviato un **link** per lo scaricamento del file **.zip** del programma. Eseguite poi il file da una cartella invece di aggiungere il programma al sistema. Dato che **Art of Illusion** è un'applicazione **Java**, dovete avere **Java Runtime Environment** funzionante nella vostra distribuzione prima di installarlo. Per ottenere la versione più recente, nei nostri test lo abbiamo scaricato dal sito e abbiamo eseguito lo **script** di installazione. Per **Dust3D**, abbiamo fatto il **download** dell'ultima **AppImage** dal sito, su cui è disponibile anche il codice sorgente per chi vuole compilarselo. Abbiamo installato la versione per **Ubuntu** di **Bforartists** scaricando dal sito il file **.deb**. Dato che è un **fork** di **Blender**, a volte è leggermente indietro rispetto all'ultima **release** di quest'ultimo in termini di funzioni e miglioramenti. Il numero di versione di **Goxel** per Linux era allineato a quello degli altri sistemi operativi, ma abbiamo dovuto fare il download del **binario eseguibile**, dopo averlo cercato un po', dalla pagina dedicata alle edizioni precedenti perché non era presente in quella principale. Il binario deposita un file **INI** nella cartella corrente. Sono disponibili anche i sorgenti da compilare e versioni **Android** e **iOS**.

Asset Forge non è Open Source e va comprato dal sito Internet del produttore

VERDETTO

BFORARTISTS	**8/10**	**DUST3D**	**7/10**
GOXEL	**7/10**	**ASSET FORGE**	**5/10**
ART OF ILLUSION	**7/10**		

Asset Forge ha un prezzo ragionevole, ma gli altri sono gratuiti. Abbiamo dovuto fare scaricamento e installazione manuale per tutti.

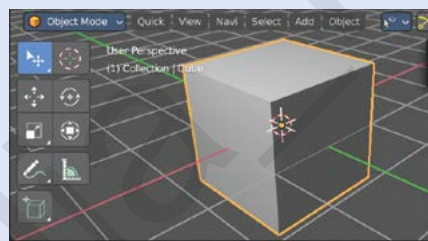
@Italia_news

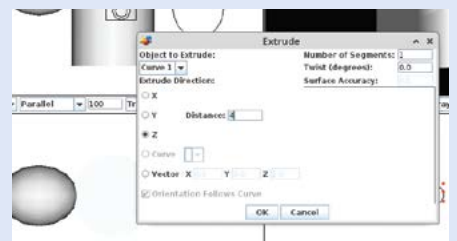
Interfaccia utente

La chiarezza è fondamentale

Sia la piacevolezza sia la produttività nella creazione di oggetti 3D dipendono dall'**usabilità** dell'interfaccia grafica. Tutti i programmi per il 3D richiedono un certo periodo di apprendimento e una **GUI** intuitiva aiuta molto. Detto questo, molto dipende anche dal gusto personale dell'utente. **Art of Illusion**, per esempio, ha un metodo di lavoro più tradizionale che può risultare rassicurante per alcuni e un po' datato per altri. È però più facile capire come funziona a prima vista di quanto non lo sia con alcune applicazioni per la grafica tridimensionale di grande successo. Quelle più moderne risultano d'altro canto più veloci sul lungo termine, quando imparate a usarle. Per i voti abbiamo considerato come criteri principali l'intuitività dell'interfaccia e la sua praticità nel consentirvi di svolgere i lavori più comuni.

Bforartists 7/10

In confronto all'estrema sobrietà degli ultimi **Blender**, l'uso del colore nelle barre degli strumenti è piacevole. Sono state eliminate alcune icone per togliere funzioni duplicate nell'**UI**, mentre ne sono state aggiunte altre per i compiti più comuni. Ci sono per esempio pulsanti dedicati per aggiungere **primitive** come cubi e cilindri nella parte alta della finestra. Anche l'uso del mouse è stato ripensato. Sono stati **deprecati** i **menu contestuali** accessibili con il tasto destro che è invece stato riservato per la rotazione. Anche se ci sono differenze, la potenza delle funzioni di Blender è rimasta e un veterano di questo programma saprebbe orientarsi bene in Bforartists, con un minimo di pratica. Tornare in Blender dopo averlo usato lo fa sembrare più legnoso, il che significa che questa interfaccia è un successo!

Art of Illusion 6/10

Art of Illusion ha un **look** molto tradizionale. Presenta una schermata con quattro pannelli, tipica dei programmi 3D meno recenti. Ci sono inoltre **finestre pop-up** e di dialogo. La funzione di **estrusione**, per esempio, attiva una finestra pop-up e potete vedere il risultato solo facendo click su **OK**. Alcuni utenti possono preferire questo approccio alla tipica finestra singola moderna. Ad aumentare ulteriormente l'intuitività, ci sono anche pulsanti per creare le principali **primitive** geometriche, come cubi e sfere, oltre a curve e poligoni. La scelta della tipologia di interfaccia, più tradizionale o moderna, è puramente soggettiva, ma l'utente medio probabilmente può orientarsi più facilmente in quella di Art of Illusion che in quelle di altri programmi.

Documentazione e community

Potete ottenere aiuto e supporto quando vi serve?

Bforartists offre dei buoni video introduttivi che sono stati creati dallo stesso sviluppatore. Coprono le basi dell'utilizzo del programma e le differenze tra la sua interfaccia e quella di **Blender**. Il principale punto di forza di questo software è infatti di essere derivato dal leader di mercato **Blender**. Grazie a questo, la maggior parte dei **tutorial** per Blender può essere adattata all'uso con Bforartists, con un po' di sperimentazione. **Art of Illusion** è in circolazione da molto tempo e ha sul sito dei buoni tutorial che vi guidano passo a passo alla realizzazione di alcuni piccoli progetti. Il **forum** non è molto vivace, ma potete ottenere rapidamente risposta alle vostre domande. Ci sono anche delle guide video su **YouTube**. Molti utenti le hanno create anche per **Asset Forge**, che ha anche un forum moderatamente attivo e un semplice manuale sul sito. Il **server Discord**, inoltre, è molto vivace ed è il posto migliore in cui andare se avete delle domande. **Goxel** offre dei video introduttivi su YouTube, ma la documentazione **online** in genere è scarsa e non c'è un forum ufficiale. Non è però un programma molto complesso da usare e questi ostacoli sono superabili. Ci sono dei tutorial su YouTube che spiegano come usare **Dust3D**. Il sito degli sviluppatori contiene inoltre un manuale online. La **community** su Internet è modesta, ma è un progetto relativamente giovane. Come nel caso di **Goxel**, non è un programma molto vasto e intricato, quindi la carenza di supporto non è un problema insormontabile, anche grazie alla presenza di esempi integrati.

VERDETTO

BFORARTISTS	8/10	DUST3D	6/10
GOXEL	5/10	ASSET FORGE	9/10
ART OF ILLUSION	7/10		

Asset Forge ha delle ottime risorse online e Bforartists può sfruttare la gran quantità di informazioni disponibili su Blender.

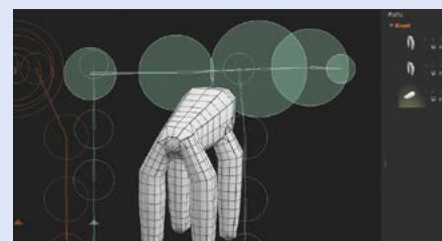
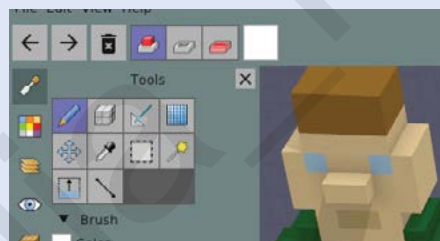
Asset Forge 8/10

Asset Forge ha l'interfaccia più semplice tra quelle delle esaminate. Fa molto bene il suo lavoro e la maggior parte degli utenti intuirà subito come funziona. Sul lato destro dello schermo c'è una serie di componenti, chiamati **widget**, che dovete posizionare per creare oggetti completi. Dopo averne aggiunto uno, potete spostarlo facendoci click sopra, oppure ruotarlo o ridimensionarlo ed è tutto molto facile. Quando mettete un widget a volte si posiziona automaticamente in modo errato e dovete spostarlo in seguito, ma non è un grosso problema. Ci sono anche delle schede che danno accesso a funzioni più avanzate come cambiare i materiali. Nel complesso, creare oggetti in **Asset Forge** è un modo intuitivo e divertente per esplorare il reame del 3D.

Goxel 7/10

Goxel è un editor di **voxel**, quindi il vostro compito principale quando modificate un oggetto o una scena è dipingere nello spazio tridimensionale. Sfortunatamente non è sempre chiaro dove si punta e a volte bisogna procedere per tentativi per posizionare un blocco. Spesso dovete ruotare la scena per accedere all'area che vi serve o inserire dei blocchi in più e cancellarli in seguito. Ci sono anche varie raccolte di strumenti con opzioni suddivise per schede nella parte sinistra della finestra. Per essere un programma abbastanza semplice, nelle schede sulla barra laterale offre molte funzioni da scoprire, come quelle per cambiare materiali e colori o alterare l'illuminazione. Nel complesso il flusso di lavoro è abbastanza lineare e creare oggetti con i blocchi è molto divertente.

Dust3D 6/10

L'interfaccia di **Dust3D** è molto ordinata ma non particolarmente comoda a volte. Sul lato sinistro dello schermo c'è una succinta barra degli strumenti. Sul lato destro c'è una serie di pannelli con schede. Gran parte dell'interfaccia, sfortunatamente, è molto piccola e scura e non c'è modo di ridimensionarla. Per usare il programma dovete avere un'ottima vista, un monitor grande o entrambi. La visualizzazione dell'area di lavoro è insolita. A differenza che in altri strumenti, la vista frontale e laterale dello scheletro della figura e la sua anteprima solida in 3D condividono la stessa area anziché avere finestre separate. Nella pratica però non è scomodo e dopo qualche minuto di sperimentazione con **Dust3D** potete iniziare a creare bizzarre creature.

Formati di importazione/esportazione

Quanto è in grado di interagire con altri programmi e motori?

Dust3D, **Asset Forge** e **Goxel** sono strumenti ideali per creare oggetti da utilizzare in altri programmi. Detto questo, Asset Forge e Goxel contengono dei propri **motori di rendering**, il che significa che possono essere usati anche autonomamente. Goxel può esportare nel comune formato **.obj**, ma perde i colori. Siamo riusciti a importare i suoi oggetti in **Blender**, a colori, usando il formato **glTF**. Il motore di gioco **Unity** non lo supporta, quindi abbiamo importato il modello in Blender e lo abbiamo esportato da lì in un formato a colori, ma questa procedura ha ovviamente comportato del lavoro aggiuntivo. Asset Forge è risultato meno laborioso perché ha delle impostazioni predefinite specifiche per Blender, per vari motori di gioco e per la stampa 3D, oltre a numerose altre opzioni di esportazione. Abbiamo provato a importare i suoi modelli in Blender e Unity e non abbiamo avuto problemi. **Dust3D** è pensato per creare oggetti da usare in altro software e abbiamo esportato un suo modello per importarlo in Blender. Il formato glTF è stato quello che ha dato i migliori risultati. Anche l'armatura per il movimento ha funzionato e abbiamo animato gli arti della creatura in Blender. **Art of Illusion** e **Bforartists** sono applicazioni complete per il rendering e la modellazione. Il secondo ha delle ottime opzioni di esportazione e importazione e, essendo un derivato di Blender, ha il beneficio di essere considerato e testato per l'importazione. **Art of Illusion** supporta meno formati ma gestisce quello .obj, che risulta il più importante all'atto pratico.

VERDETTO

BFORARTISTS	9/10	DUST3D	7/10
GOXEL	7/10	ASSET FORGE	9/10
ART OF ILLUSION	6/10		

Asset Forge ha funzionato molto bene e offre tutte le opzioni di esportazione che possono servire. Anche Bforartists si è dimostrato valido.

Rendering

Qual è la qualità dell'output?

Bforartists è analogo al leader di mercato **Blender** sotto questo profilo. Integra infatti più di un **motore di rendering** e può lavorare anche con vari esterni, sia Open Source sia commerciali (come **RenderMan** di **Pixar**). Usando il **renderer** integrato **Cycles** può sfruttare l'**accelerazione GPU** con risultati potenzialmente cinematografici. **Art of Illusion** ha un motore di rendering integrato simile a esso che usa il **ray tracing**. È basato sul software ma molto veloce. Potete trovare dei bellissimi esempi dei risultati sul sito del progetto, ma non è allo stesso livello di Blender. **Dust3D** non integra il supporto per il rendering ma questo non è un difetto: semplicemente non è tra gli scopi del programma. **Goxel** può creare delle immagini basate sull'aspetto della scena nella **viewport** e sono di buona qualità. Ha anche un semplice motore di rendering con ray tracing integrato che però è lento in quanto basato sul software. La soluzione più semplice è esportare immagini o modelli per un rendering dettagliato in un altro programma. Anche **Asset Forge** può renderizzare le immagini con il motore integrato. L'esportazione corrisponde a quanto visualizzato a schermo e, dal punto di vista estetico, i risultati sono sempre definiti e brillanti. Le opzioni di rendering sono limitate ma nel complesso fanno un buon lavoro.

Bforartists spicca nel **rendering** rispetto agli altri programmi del confronto

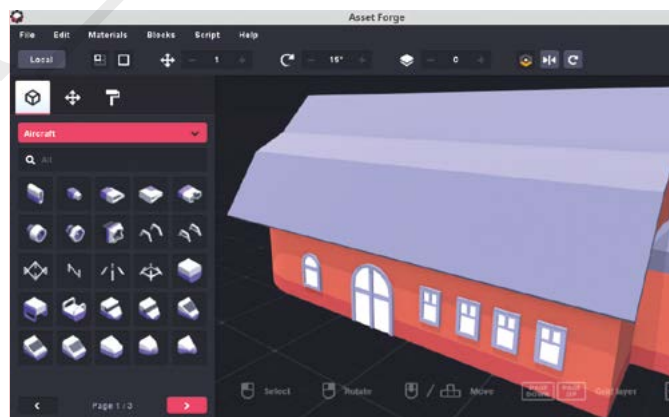
VERDETTO

BFORARTISTS	10/10	DUST3D	N/D
GOXEL	7/10	ASSET FORGE	7/10
ART OF ILLUSION	7/10		

Bforartists spicca su tutti per la potenza nel rendering. Anche Goxel e Asset Forge possono dare autonomamente un buon output.

Facilità d'uso

Che impegno richiedono?

Nel considerare la facilità d'uso, tenete presente che una curva di apprendimento più ripida è spesso giustificata da un maggior numero di funzioni. Allo stesso modo, un'applicazione può essere semplice da usare ma non riuscire a svolgere i compiti che vi servono. **Bforartists** fa un buon lavoro nel semplificare l'esperienza di **Blender**. L'uso di più colori è piacevole e rende l'interfaccia più facile da navigare. Ci sono anche molte più icone, varie delle quali compaiono quando servono, il che rende più ordinato lo schermo. Il suo cuore è però sempre Blender quindi se volete andare oltre le basi dovete studiare un po'. Iniziare a lavorare in **Dust3D** è abbastanza facile e risultano molto utili i bei file di esempio integrati, che vi aiutano a capire subito come funziona il programma. **Art of Illusion** è un software completo per il **modelling**, l'animazione e il **rendering** in 3D e di conseguenza non può che richiedere un minimo di studio per consentire all'utente di svolgere funzioni più avanzate che semplicemente aggiungere delle forme geometriche a una scena. Non ha però la stessa complessità di altri prodotti analoghi e l'interfaccia utente facilita il processo di apprendimento. **Asset Forge** risulta molto accessibile e probabilmente capirete come funziona semplicemente vedendolo in esecuzione. Basta fare click su un componente, posizionarlo nella finestra principale e poi spostarlo, ruotarlo e ridimensionarlo in base alle proprie esigenze. Le funzioni avanzate includono la possibilità di cambiare i materiali e lo schema cromatico di un oggetto.

Non è difficile intuire le modalità di funzionamento di **Asset Forge** semplicemente osservando lo schermo una volta che i componenti **3D** sono visualizzati

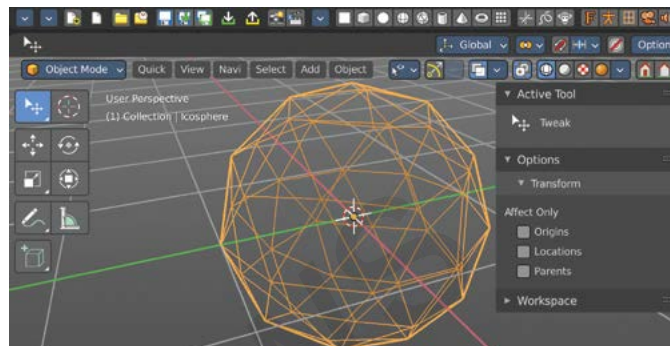
I controlli da tastiera vengono sempre mostrati alla base dello schermo, il che è pratico. Potete iniziare a lavorare con **Goxel** quasi immediatamente: **zoomate** con la rotella del mouse e iniziate a posizionare i blocchi sulla vostra scena. Imparate poi a cancellarli e colorarli. I dieci pannelli che trovate sulla barra degli strumenti a sinistra vi offrono inoltre molte altre opzioni da esplorare.

VERDETTO

BFORARTISTS	7/10	DUST3D	7/10
GOXEL	8/10	ASSET FORGE	9/10
ART OF ILLUSION	8/10		

Asset Forge e Goxel sono un piacere da usare, ma non sono applicazioni con moltissime opzioni e funzioni avanzate.

Il verdetto

Tutti gli strumenti per il 3D esaminati in questo articolo sono validi e utili. Il principale criterio di giudizio è stata la facilità d'uso, ma sono state prese in considerazione anche le capacità del programma e, per questo motivo, **Bforartists** ha ottenuto il punteggio più alto. Offre infatti tutte le funzioni di **Blender**, ma lo rende più facile da gestire sia nell'uso quotidiano sia per chi non conosce ancora l'argomento. Ha eliminato l'eccesso di menu e icone, spesso duplicati, del programma da cui deriva, e ogni aspetto del suo utilizzo è stato migliorato in qualche modo. L'ampio spettro di opzioni offerte dall'applicazione comporta però un'inevitabile complessità di fondo e per imparare a usarlo al di là delle funzioni di base è necessario studiarlo. **Art of Illusion** è un programma completo e molto accessibile. Per quanto il suo semplice approccio alla modellazione e al **rendering** siano molto apprezzabili, però, le sue capacità sono relativamente limitate e le dimensioni della sua **community** non sono paragonabili a quelle offerte da Blender. Potrebbe d'altro canto essere utile in ambito didattico, dove il tempo a disposizione è limitato, e offre anche delle funzioni moderne e avanzate come la **modellazione procedurale** (con un **editor dei nodi**). Non supera però Blender e i suoi derivati in nessuna area specifica, da un punto di vista tecnico. Il progetto è attivo da oltre vent'anni ed è attualmente mantenuto. **Asset Forge** vi permette di creare oggetti 3D a partire da componenti singoli ed è divertente da usare e utile per una rapida creazione di modelli. I componenti inclusi permettono però di produrre un numero limitato di tipologie di oggetti, come costruzioni e veicoli. Potete ampliare le potenzialità del programma aggiungendo altri componenti, ma questo va a penalizzarne la semplicità complessiva. Come spesso capita con i programmi commerciali, l'interfaccia utente è molto fluida e ben curata. **Goxel** è un altro strumento molto piacevole da usare e, se vi piace lo stile grafico che si può ottenere utilizzando i **voxel**, potete usarlo per creare degli ottimi modelli. C'è anche una versione per il Web. **Dust3D** è un progetto ancora relativamente giovane, con qualche elemento non perfettamente definito nell'interfaccia. Ci sono degli aspetti da studiare ma in un paio d'ore potrete realizzare le vostre creature e iniziarne a muovere gli arti. LXP

1° Bforartists 9/10

Web: **www.bforartists.de** **Licenza:** **GPL**
Versione: **2.9.3 (basato su Blender 3 alpha)** Offre tutte le funzioni di Blender, ma con un'interfaccia più facile da usare.

2° Art of Illusion 8/10

Web: **www.artofillusion.org** **Licenza:** **GPL 2**
Versione: **3.2.0** È un programma completo per la modellazione in 3D, ma più intuitivo e rapido da imparare di altri strumenti in questo articolo.

3° Asset Forge 7/10

Web: **https://assetforge.io** **Licenza:** **Commerciale**
Versione: **2.2.0** È divertente da usare e vi permette di creare dei bei modelli. Risulta però un po' limitato nelle applicazioni.

4° Goxel 7/10

Web: **https://goxel.xyz** **Licenza:** **Doppia – GPL-3.0 o commerciale** **Versione:** **0.10.8** Il look dei voxel non è mai stato più popolare e questo strumento è ottimo per produrre modelli in quello stile.

5° Dust3D 7/10

Web: **https://dust3d.org** **Licenza:** **MIT**
Versione: **1.0.0-rc.6** Serve un po' di tempo per orientarsi ma poi potete realizzare facilmente creature con arti in movimento.

» CONSIDERATE ANCHE QUESTE ALTERNATIVE

Lo sviluppatore di **Asset Forge** offre anche l'**editor di voxel** economico **KenShape** (**https://tools.kenney.nl/kenshape**). È divertente per giocarci un po', ma limitato dal fatto che tutti gli oggetti devono essere simmetrici almeno su un asse. **Zoxel** (**http://zoxel.blogspot.com**) è un'altra applicazione Open Source per la modellazione con i voxel. **Gloss3D** (**http://gloss3d.net**) e **Wings3D** (**www.wings3d.com**) sono due programmi Open Source completi per il **modelling** 3D. Il modulo di disegno di **LibreOffice** (**www.libreoffice.org**) ha delle capacità di base per la grafica tridimensione. Potete aggiungere oggetti 3D al disegno e manipolarli, per esempio per creare testo tridimensione. **FreeCAD** (**www.freecadweb.org**) è un'altra opzione popolare per la grafica 3D in **Linux**.

Da non perdere

Audacity » Clapper » Haguichi » Media Downloader » Panon » Radiotray-NG » Dragit » Balla » CommandLineHeroes

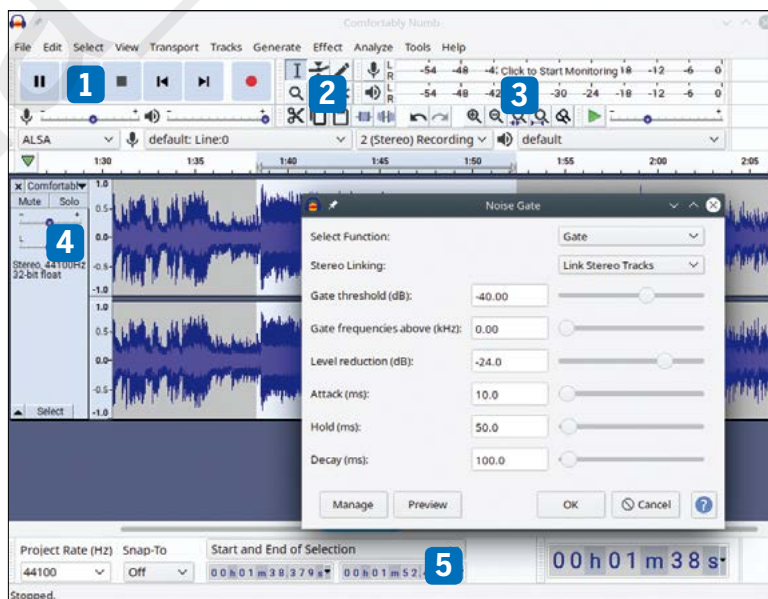
EDITOR AUDIO

Audacity

Versione: 3.03
Web: www.audacityteam.org

Audacity è un apprezzatissimo **editor audio multitraccia** che permette, per esempio, di creare un **podcast** combinando in un unico progetto diverse tracce (come la voce e la musica di sottofondo), ridurre il rumore, applicare riverbero e altri effetti e molto altro. Si tratta di un'applicazione Open Source di alto livello adatta sia agli amatori sia ai professionisti. Una delle sue funzioni permette di selezionare manualmente i dispositivi audio di **input** e **output** permettendovi di usare **ALSA** al posto di **PulseAudio** se necessario. L'interfaccia contiene una barra degli strumenti principali in alto con l'area delle tracce sotto di essa. L'impatto è chiaro e ordinato e ci sono moltissimi strumenti, effetti e **plug-in** per l'elaborazione audio. L'interfaccia grafica è basata su **wxWidgets** e **GTK**. La **release 3.0** ha aggiunto una serie di funzioni, tra cui il nuovo formato interno **AUP3** per salvare i progetti, che utilizza un **database SQLite3** per migliorare le prestazioni di creazioni multitraccia avanzate e offre tempi di accesso prevedibili quando si gestiscono risorse molto lunghe, a prescindere da quale dispositivo di salvataggio venga usato. Con AUP3, inoltre, i dati del progetto vengono salvati in un singolo file, il che è un notevole passo avanti rispetto all'insieme di file, piuttosto confuso, che veniva utilizzato nelle precedenti versioni. Tra le altre novità, il filtro per il rumore **Noise Gate** è ora molto più preciso ed efficiente e il nuovo strumento **Label Sounds** vi permette di aggiungere **tag** a parti del progetto. La release **3.0.1** era mirata alla migliore identificazione dei **bug** e rivolta gli sviluppatori, la **3.0.2** ha portato alla risoluzione di molti problemi e la **3.0.3**, oltre a sistemare altri bug, ha introdotto varie migliorie tra cui il fatto che l'effetto **Equalization** è ora diviso in due (**Filter Curve** e **Graphic EQ**). Le esportazioni sono inoltre più precise e mantengono i silenzi iniziali.

Le nuove versioni di **Audacity** dalla 3.0.0 portano miglioramenti e nuove funzioni

L'INTERFACCIA DI AUDACITY

1 Controlli di riproduzione Usate questi pulsanti per riprodurre e interrompere i file del progetto e anche per lanciare la registrazione.

2 Strumenti della forma d'onda Potete copiare, tagliare e incollare parti della forma d'onda e persino "disegnarci sopra" usando questi strumenti.

3 Area di monitoraggio Qui potete monitorare la riproduzione o la registrazione, ingrandire o ridurre la barra di misurazione e controllare i tempi.

4 Regolazioni per la singola traccia In un progetto possono esserci più tracce. Per ciascuna potete cambiare volume, bilanciamento sinistra/destra e molto altro.

5 Selezioni di precisione Il pannello in basso ha un'area dedicata per modificare la selezione della forma d'onda, anche di una frazione di secondo.

RIPRODUTTORE VIDEO

Clapper

Versione: 0.3.0 **Web:** https://github.com/Rafostar/clapper

Clapper a prima vista è solo un'applicazione **GTK** minimalista con un look accattivante. Apre tutti i video supportati dai **codec Gstreamer** e li riproduce. Fin qui niente di nuovo, ma ci sono tanti piccoli dettagli che permettono al programma di farsi notare. Clapper è infatti un'applicazione **GTK4 next-gen** che renderizza tutto, sia il video sia l'interfaccia grafica, attraverso **OpenGL**. Offre anche l'**accelerazione VAAPI** preimpostata (date un'occhiata al vostro monitor di sistema per vedere com'è basso l'utilizzo della **CPU**) e un controllo del volume con sovra-amplificazione. Permette anche di cambiare facilmente la lingua dei sottotitoli dalla barra di controllo principale e offre una **modalità fluttuante**. Come dichiara il suo autore, il programma usa la versione **patchata** del componente **GstPlayer** e questo offre dei benefici in termini di prestazioni.

Clapper ha un pannello di controllo dedicato per la modalità a schermo intero

Il tempo di ricerca per i video ad alta risoluzione o molto lunghi è eccezionalmente breve, per esempio. Se **Gstreamer** nel vostro sistema ha già queste **patch**, provate la versione **Flatpak** di Clapper. Ci sono già molti riproduttori multimediali sul mercato ma questo (creato dall'autore chiedendo su **Reddit** quali sarebbero state le funzioni migliori da mettere nel **video player** perfetto) merita un giro di prova. È un progetto giovane e promettente che è basato sulle reali esigenze degli utenti. Nei nostri test si è già rivelato più potente e robusto **Totem**, che è il riproduttore video predefinito di **Gnome**.

CLIENT HAMACHI

Haguichi

Versione: 1.4.4 **Web:** https://github.com/ztefn/haguichi

Haguichi è un **client** grafico moderno per **Hamachi**, uno strumento per la creazione e la gestione di **VPN** che permette di configurare reti locali su Internet. Potete ottenere moltissimi benefici dal creare una rete **LAN** composta da vari computer collegati alla rete senza aver bisogno di riconfigurare **router**, **NAT** e **firewall**, o di impostare il **port forwarding** e altri compiti correlati. Hamachi è di per sé un software proprietario, con una versione Linux ufficiale che consiste solo di un'interfaccia a riga di comando, senza versione grafica. Haguichi è invece un'applicazione Open Source che offre un'interfaccia **GTK3** per creare reti Hamachi o unirsi a esse. Nei nostri test abbiamo impiegato un paio di minuti per rendere operativo il suo codice, dato che il programma ha un numero limitato di **dipendenze** per la compilazione (tra cui **Vala**, **GTK** e **libnotify**). Per poterlo usare, però, dovete avere Hamachi in esecuzione. Ha un suo servizio **Systemd** che attiva un'interfaccia di rete

Gestite facilmente le reti **Hamachi** con questa applicazione pratica e robusta

separata (di solito **ham0**). Haguichi apre una finestra compatta e mostra subito gli indirizzi di rete (sia **IPv4** sia **IPv6**) della vostra nuova LAN. Premete il tasto **+** per creare una rete o unirvi a una esistente. Online ci sono elenchi di reti Hamachi pubblicamente disponibili, ma sono quasi tutti creati da comunità di giocatori e diventano rapidamente obsoleti. Se lo avete installato su almeno un paio di macchine, vale la pena di provare questa applicazione Open Source. Create un **login** Hamachi, registratevi e impostate una nuova rete (`$ hamachi create <nome rete> <password>`). Collegatela quindi a Haguichi usando lo stesso nome utente e password e siete pronti a usarlo. Il programma vi permette di **pingare** i membri della LAN, accedere alle loro condivisioni **Samba** e unirvi a più reti alla volta.

SUPPORTO AGLI SCARICAMENTI

Media Downloader

Versione: 1.6.0 **Web:** https://github.com/mhogomchungu/media-downloader

Aggiungete ulteriori opzioni per i vostri scaricamenti e monitoratene l'avanzamento

Ci sono molti **front-end** per l'**utility youtube-dl**, ma non tutti funzionano correttamente o sono ancora mantenuti. **Media Downloader** è un'applicazione grafica multischeda che vi dà accesso sia a **youtube-dl** sia a **wget**. Per i **download** standard potete usare quest'ultimo senza lanciare una finestra dal terminale ma lo scopo principale dell'applicazione è aiutarvi a scaricare video e salvarli localmente, non solo da **YouTube** ma da centinaia di servizi video, siti di canali televisivi, portali per l'intrattenimento e altro. L'interfaccia del programma è chiara e facile da usare ed è basata sulle schede **Basic**, **Batch** e **Playlist**. Nel primo caso, basta fornire l'**URL** per far scaricare il video nella **cartella Home**. Batch implica che abbiate una serie di URL pronti da aggiungere alla lista di Media Downloader. Potete anche fornire un file di testo esterno con i collegamenti, il che è utile se volete scaricare molti video. La sezione Playlist, infine, estrae e scarica video dal link a una playlist su YouTube. Nei nostri test ha funzionato correttamente. Tra le funzioni offerte ci sono anche il ridimensionamento per i **display HiDPI** e, naturalmente, la possibilità di cambiare la cartella in cui salvare i video. Nell'insieme è uno strumento pratico e affidabile che vale la pena di provare.

WIDGET PLASMA5

Panon

Versione: 0.4.6
Web: https://github.com/rbn42/panon

Dalle impostazioni di **Panon** potete anche cambiare **backend** audio e regolare la latenza

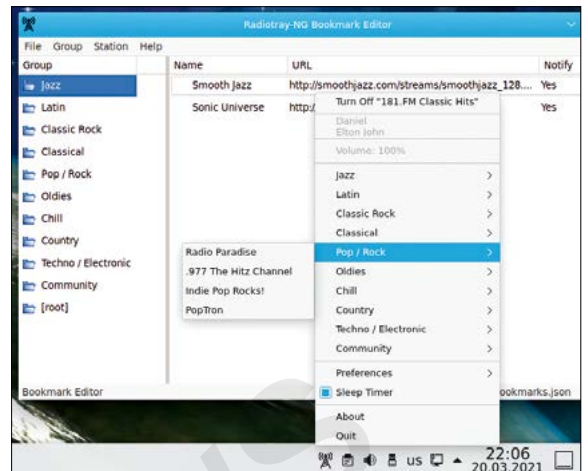
Panon applica effetti visivi a ogni suono riprodotto sul vostro computer con **Linux**. Parte da uno **spettrogramma** abbastanza semplice del segnale audio per arrivare a effetti più sofisticati come tunnel, immagini astratte tipo **venti solari**, mesmerizzanti **frattali di Mandelbrot** e molto altro. Avrete sicuramente visto almeno alcuni di questi effetti in altri programmi multimediali, come per esempio **Totem** di **Gnome** e **Windows Media Player**. Panon è però diverso da questi strumenti per vari motivi. Prima di tutto ci sono moltissimi effetti inclusi e ancora più opzioni disponibili per lo scaricamento direttamente dal programma. Panon è inoltre un **widget** per **Plasma5** che si può usare all'interno di un pannello o un **dock**, oppure direttamente sul **desktop KDE Plasma**. Oltre a creare dei begli effetti grafici, infine, è anche utile perché disegna uno spettrogramma colorato ma realistico. Provatelo con le vostre registrazioni di stili diversi; traccia il corretto spettro di frequenza, con i bassi sulla sinistra, i medi al centro e gli alti sulla destra. Il catalogo di effetti del programma offre varie opzioni per lo spettro e alcune di esse permettono di personalizzare i colori, l'impatto degli effetti grafici e altro. Potete usare questo **widget** in vari modi per rendere la vostra scrivania con Plasma5 più vibrante, per esempio creando un pannello in alto occupato da Panon e poi un **overlay** con un pannello **Latte** delle stesse dimensioni con i vostri widget preferiti (**Menu, Tray**, ecc.). Rendete sfocato e semitrasparente il secondo pannello e, ogni volta che riprodurrete qualcosa, pulserà a ritmo di musica. Panon sfrutta l'**accelerazione OpenGL** ed è disponibile sul **KDE Store**, da browser o con **KGetNewStuff** in Plasma5.

@italia_news

SINTONIZZATORE RADIO

Radiotray-NG

Versione: 0.2.7
Web: https://github.com/ebruck/radiotray-ng

Nella maggior parte dei casi, i riproduttori musicali inclusi nelle distribuzioni sono poco utili se non avete una vostra libreria di brani audio. Alcuni, come **Clementine**, sono collegati a **provider online** di musica, mentre altri offrono una selezione di radio (**Elisa** e **Rhythmbox**). Di solito è però limitata e non risponde alle esigenze di tutti. **Radiotray-NG** risolve il problema fornendo una buona gamma di stazioni radio che coprono i gusti della maggior parte degli utenti. Il programma offre un'icona per la **barra delle applicazioni** (basata su **libappindicator**) che si espande in una lista di stazioni divise per categoria. Propone inoltre una serie di funzioni extra di solito assenti in questo tipo di strumento. Ci sono un **timer** per chiudere automaticamente le stazioni inattive, un controllo del volume che regolate scorrendo con la rotella del mouse sull'icona di **Radiotray-NG**, i dettagli delle tracce correnti e naturalmente l'**editor** integrato delle stazioni.
Il riproduttore supporta i gruppi e salva le informazioni in formato **JSON**. L'interfaccia è basata su **GTK3** e usa **Gstreamer** come **backend** audio. Installare Radiotray-NG vi permette di avere più fonti musicali in **Linux**, ma ci sono dei limiti al supporto delle icone per la barra delle applicazioni in generale. Al momento di scrivere l'articolo, per esempio, non si può accedere al programma in **Gnome 40** perché manca l'estensione **Top Icons** (o un suo equivalente). Se il vostro **ambiente desktop** non ha problemi di questo tipo, però, Radiotray-NG è un ottimo

Con **Radiotray-NG** potete esplorare stazioni radio di vari generi

CONDIVISIONE FILE

Dragit

Versione: 0.6.0 **Web:** https://github.com/sireliah/dragit

Condividere file in rete con **Linux** è diverso che in **Windows** o **macOS**, non in termini di velocità di trasferimento ma nelle modalità operative nei vari ambienti desktop disponibili. Entrare nelle proprietà di un file o di una cartella e selezionare qualche riquadro nella scheda **Share** non lo rende immediatamente condiviso, per via di questioni con **Samba** o altri tecnicismi. Per questo ci sono vari strumenti di condivisione di terze parti. Nei nostri test con **Dragit**, sono emerse delle forti somiglianze con un'applicazione analoga di nome **Dukto**. Quest'ultima, però, non viene aggiornata da tempo e, se vi serve uno strumento per la condivisione di file, conviene affidarsi a Dragit. Si tratta di un'applicazione **stand-alone** scritta in **Rust** e **GTK**, che sfrutta la libreria **libp2p** e usa il protocollo **Multicast-DNS** per collegarsi ad altri dispositivi nella rete locale. Dovete prima installare i programmi sui due **device** usati per il trasferimento, verificano che il **firewall** non interferisca. Controllate quindi che la vostra istanza di Dragit abbia rilevato il destinatario e semplicemente trascinate un file sul suo nome. Non potete condividere in questo modo una cartella o più file contemporaneamente, ma si può trascinare una selezione di testo, come se ci fossero degli appunti condivisi tra i due dispositivi. Potete cambiare la **directory** di salvataggio per i file ricevuti e anche visualizzare l'avanzamento del trasferimento usando i **pop-up** di notifica del programma. Attualmente **Dragit** supporta **host** Linux e Windows ed è pensato soprattutto per lo scambio di file tra **desktop** e **workstation**.
Offre un modo semplice e pratico per passare file da una macchina all'altra senza bisogno di fare sostanzialmente alcuna configurazione.

Trascinate un file o del testo sulla finestra di **Dragit** per trasferirlo facilmente

GIOCO DI ABILITÀ

Balla

Versione: GIT **Web:** https://github.com/TimothyGramnaes/Ball-game

Balla è un progetto in **JavaScript** creato da un gruppo di giovani sviluppatori svedesi e mescola elementi di biliardo e **puzzle game**. Potete accedervi dal **browser**, sia da computer sia da un dispositivo mobile. L'area di gioco contiene tre elementi principali: un grosso bersaglio rosa, una piccola palla gialla e la palla lilla che dovete spostare con il mouse o con un dito. Lo scopo è colpire la palla gialla con quella lilla in modo da farla rimbalzare contro il bersaglio, facendolo rimpicciolire. Dopo ogni collisione il bersaglio rilascia delle specie di coriandoli e la palla viola deve evitare queste particelle. Il tutto è divertente, ma piuttosto difficile, soprattutto le prime volte che giocate. L'obiettivo è aumentare il livello di attenzione dell'utente e la sua capacità di controllare contemporaneamente più oggetti in movimento. Nell'angolo in alto a destra c'è una barra con dei cuori che indicano la vostra salute (potete sbagliare solo tre volte) e in quello a sinistra c'è un timer. Per vincere la partita dovete distruggere il bersaglio completamente, il che diventa difficile verso la fine perché più piccolo diventa, più veloce si muove. Nel frattempo la quantità di coriandoli può essere diminuita dalla palla gialla, che li "mangia". È però un aspetto rischioso perché vi porta a cercare di colpire attivamente la palla gialla invece di stare nella parte periferica della schermata. L'idea di Balla è divertente, la grafica è bella e l'implementazione è di qualità. Anche la colonna sonora è piacevole!

Non fatevi sviare dai colori pastello: **Balla** offre un'esperienza di gioco intensa!

GIOCO ARCADE

CommandLineHeroes

Versione: GIT **Web:** https://github.com/CommandLineHeroes/clh-bash

Conoscete davvero molti comandi di **Linux**? Mettetevi alla prova con questo divertente gioco sul Web supportato da **Red Hat**! **CommandLineHeroes** (o **CLH**) è un **gioco arcade** veramente originale. Trasforma il terminale in una macchina da gioco in cui il vostro obiettivo è scrivere il maggior numero possibile di comandi validi entro un minuto. Sono accettati quelli di **Bash**, **JavaScript**, **Python** e **HTML5**. Naturalmente potete provare a barare, ma l'assenza della funzione copia e incolla nel terminale virtuale di CLH e il tempo limitato a vostra disposizione lo rendono faticoso. L'idea è semplice ma il progetto è professionale e ben realizzato. Il merito va in gran parte al motore **Hero** basato su **Tiled** e **Phaser**. Dopo la fase di impostazione iniziale (di cui trovate i dettagli in **Readme.md**), ricevete l'**URL** del gioco e potete cominciare la sfida. Scrivete **play** e un'animazione vi porta rapidamente davanti a un computer immaginario in stile retro. Il gioco mostra una breve presentazione testuale e le regole per i punteggi e vi prepara per l'impresa. Dopo che avete fatto del vostro meglio nei sessanta secondi a vostra disposizione, CLH visualizza un punteggio dettagliato che include il conteggio dei comandi validi inseriti per ciascun linguaggio e poi vi porta alla **tabella dei record**. Giocare con **CommandLineHeroes** è un modo divertente per mettere alla prova la vostra memoria e abilità dalla riga di comando ma è anche una grande fonte di ispirazione per sfide tra sviluppatori o amministratori di sistema. Il database del gioco contiene quasi 3.600 comandi di Bash, 150 di HTML5, 350 di JS e 200 di Python, il che consente delle belle partite! LXP

La raccolta di comandi di **Bash** è la più ricca, ma il gioco offre anche una bella sfida anche per chi sviluppa sul Web

Tutorial

I nostri esperti offrono i loro consigli di programmazione e di amministrazione del sistema

LA GUIDA DI RIFERIMENTO

Esiste sempre qualcosa di nuovo da imparare in campo informatico, soprattutto in un mondo dinamico come quello di Linux e dell'Open Source. Ogni numero di Linux Pro presenta una sezione dedicata a tutorial realizzati da esperti in moltissimi settori: programmazione, sicurezza, amministrazione di sistema, networking. Troverete informazioni utili sia che siate dei veterani di Linux sia degli utenti alle prime armi. Studieremo con cura anche le applicazioni più diffuse sia in ambito lavorativo che desktop. Il nostro scopo è quello di fornire in ogni numero il giusto mix di argomenti, ma se avete suggerimenti su temi particolari che vorreste vedere trattati, scriveteci via e-mail all'indirizzo tutorial@linuxpro.it

COME RAPPRESENTIAMO LE LINEE DI CODICE

Si presenta spesso la necessità di riportare le linee di codice di un programma. Per favorirne la lettura evidenzieremo le singole linee in questo modo:

```
begin
  mniWordWrap.Checked := not
end
```

Quando una riga di codice supera la lunghezza della colonna la riporteremo su più righe utilizzando la notazione seguente:

```
printf("Vi preghiamo di inserire
una password.");
```

TUTORIAL

ACCADEMIA DEL CODICE

SYSTEMD-ANALYZE

Avvio più veloce

A nessuno piace attendere che termini la fase di boot del computer e ci sono strategie per renderla molto veloce con tutte le distribuzioni

Oggi quasi tutti gli ambienti grafici e le applicazioni sono facilmente disponibili per la maggior parte delle distribuzioni, il che le rende meno differenziate tra loro sotto questo aspetto. Un'area, però, in cui competono accanitamente tra di loro è quella delle prestazioni e dell'uso ottimale delle risorse di sistema per avviarsi nel minor tempo possibile. Quasi tutte le principali usano **systemd** come **sistema di init**. A differenza di altri strumenti analoghi come **System V**, systemd è una **suite software** che comprende vari **demoni** e altri componenti e **utilità di sistema**. Tra questi ci sono **systemctl**, che si può usare per avviare e interrompere oppure attivare e disattivare un servizio, e **systemd-analyze** che permette di studiare nel dettaglio le statistiche delle prestazioni all'avvio. Se vi sembra che la vostra distribuzione sia troppo lenta nella fase di **start**, esaminando questi dati potete trovare i colli di bottiglia e rendere più veloce il processo. Non c'è bisogno di installare systemd-analyze perché fa parte delle applicazioni della suite di systemd. Se eseguite il comando **systemd-analyze** senza altre opzioni, lo strumento vi restituisce il tempo totale di avvio del sistema:

```
$ systemd-analyze
Startup finished in 5.656s (kernel) + 1min 6.557s (userspace) = 1min 12.214s
graphical.target reached after 1min 5.129s in userspace
```

Come potete vedere, il comando indica quanto tempo è stato necessario per avviare il **kernel**, oltre che per attivare i componenti dello **spazio utente**. Nei tempi del kernel sono inclusi anche quelli dei **driver** dei dispositivi che devono essere avviati quando aprite il sistema operativo. Tutto il resto, come per esempio le librerie o altri demoni, fa parte dello spazio utente. Dall'**output** del comando **systemd-analyze** potete vedere che sono stati necessari solo 5,6 secondi per l'avvio del kernel, ma quello dello spazio utente ha richiesto più di un minuto. Come è facile immaginare, si tratta di un tempo molto lungo, soprattutto per una macchina **quad core** con **12 GB** di **RAM**.

Il comando **systemd-analyze security** si può usare per controllare la sicurezza e le impostazioni delle **sandbox** dei vari servizi

Parte la caccia al colpevole

Per scoprire quale servizio è responsabile dei tempi di avvio, dovete eseguire il comando **systemd-analyze blame** ed esaminare ciò che restituisce:

```
$ systemd-analyze blame
```

» ELIMINARE I COLLI DI BOTTIGLIA

Dopo aver identificato i colli di bottiglia responsabili del rallentamento dei vostri tempi di avvio potete usare il comando **systemctl** per interrompere e disabilitare i servizi che non volete. Nell'esempio presentato in precedenza, **smdb**, **nmdb** e **networkmanager** sono tre dei servizi che ne tengono bloccati altri. Per interromperli e disabilitarli in modo che non vengano attivati all'avvio, eseguite i seguenti comandi:

```
$ sudo systemctl stop smdb
$ sudo systemctl disable smdb
```

Il primo comando interrompe il servizio in esecuzione al momento mentre il secondo impedisce che sia riattivato automaticamente all'avvio. Anche dopo averlo fatto, però, potreste trovarvi il colpevole in esecuzione a un successivo riavvio, se un altro dipende da esso e lo avvia. Trovate un elenco dei servizi che ne possono attivare altri, come per esempio smdb o **snapd**, con il comando **systemd-analyze blame | grep <servizio>**. **systemd-analyze blame | grep snapd** mostra, per esempio, tutti gli elementi che hanno bisogno di un servizio padre. Per vedere una lista di tutti i servizi abilitati, eseguite il comando **systemctl list-unit-files | grep enabled**.

```
28.458s udisks2.service
27.409s dev-sda7.device
23.643s snapd.service
21.156s apt-daily.service
17.497s dev-sda6.device
16.380s networkd-dispatcher.service
15.827s systemd-journal-flush.service
```

L'**output** completo del comando è piuttosto lungo e qui ne viene riproposta solo una parte per motivi di spazio. Potete navigare al suo interno con i **tasti freccia**. Nei dati restituiti i diversi servizi sono divisi in base al tempo necessario richiesto per lanciarli. Molti di essi si avviano in parallelo ed è per questo che il totale di queste cifre è significativamente più alto del tempo reale riportato da systemd-analyze. Questa **utility** vi dà un'indicazione dei servizi che possono richiedere un maggiore scrutinio se volete migliorare i **tempi di boot**. Tenete anche presente che questi ultimi variano a ogni riavvio, in base ai cambiamenti che avete fatto al sistema installando o rimuovendo applicazioni. Nel nostro esempio, **snapd.service** richiede 23,6 secondi. Smettendo di usare i **pacchetti Snap** e interrompendo oppure disabilitando il servizio relativo si migliorerebbero i tempi di avvio. Non vale però la pena di introdurre modifiche al sistema sulla sola base dei dati ottenuti con questo comando.

Capire la reazione a catena

Alcuni processi della procedura di avvio non possono iniziare fino a quando altri, da cui dipendono, non sono stati attivati. Potete ottenere una lista di questi ultimi usando il **sottocomando** `critical-chain` :

```
$ systemd-analyze critical-chain
Il momento in cui l'unità si è attivata è printato dopo
il carattere "@".
Il tempo che l'unità ha impiegato per l'avvio è printato
dopo il carattere "+".
graphical.target @57.345s
└ multi-user.target @57.344s
 └ smbd.service @53.719s +3.625s
  └ nmbd.service @43.264s +10.452s
   └ network-online.target @43.090s
    └ NetworkManager-wait-online.service @33.967s
+9.122s
     └ NetworkManager.service @27.506s +6.459s
      └ dbus.service @27.502s
       └ basic.target @27.366s
        └ sockets.target @27.366s
         └ snapd.socket @27.361s +1ms
          └ sysinit.target @27.189s
           └ swap.target @27.189s
            └ dev-disk-by\x2duuid-982f7286\
x2d84f5\x2d48e5\x2d807e\>
             └ dev-disk-by\x2duuid-982f7286\
x2d84f5\x2d48e5\x2d807>
```

Qui, i numeri dopo il simbolo @ mostrano il numero assoluto di secondi, dall'inizio della procedura di avvio, dopo cui l'unità si è attivata. Similmente, quelli dopo il carattere + indicano il tempo necessario per l'avvio dell'unità. Potete anche controllare la **critical chain** di qualsiasi servizio con il comando `systemd-analyze critical-chain <servizio>` :

```
systemd-analyze critical-chain NetworkManager.service
Il momento in cui l'unità si è attivata è printato dopo il
carattere "@".
Il tempo che l'unità ha impiegato per l'avvio è printato
dopo il carattere "+".

NetworkManager.service +6.459s
└ dbus.service @27.502s
 └ basic.target @27.366s
  └ sockets.target @27.366s
   └ snapd.socket @27.361s +1ms
    └ sysinit.target @27.189s
     └ swap.target @27.189s
      └ dev-disk-by\x2duuid-982f7286\x2d84f5\
x2d48e5\x2d807e\x2de208361fb>
       └ dev-disk-by\x2duuid-982f7286\x2d84f5\
x2d48e5\x2d807e\x2de208361>
lines 1-12/12 (END)
```

I dati generati in `systemd-analyze` si possono rappresentare in un grafico per renderli più fruibili. Eseguite il comando `systemd-analyze plot > /tmp/systemd-chart.svg` per creare un grafico in **Scalable Vector Graphics** (**.svg**) nella cartella **/tmp**. Potete scegliere di salvarlo in un'altra **directory** a vostra scelta se volete conservarlo per futuro riferimento. Il comando crea un **file SVG**, ossia un documento testuale che definisce una serie di **vettori grafici**. Potete poi usarlo con un'applicazione come **Image Viewer**, **Ristretto** o **LibreOffice Draw** per generare un grafico. L'immagine che risulta da questo processo è molto grande e mostra tutti i servizi che si sono attivati all'avvio, oltre al tempo che hanno impiegato per farlo e a tutte le loro dipendenze. Il **critical path** è evidenziato in rosso. Dovete **zoomare** considerevolmente per poter capire i dati rappresentati e, se volete sfruttare al meglio le possibilità offerte da systemd per la risoluzione dei problemi nel **system manager**, è molto utile leggere le indicazioni della sua **pagina man**. LXP

C'è una semplice tecnica per impedire che un servizio disabilitato sia lanciato all'avvio: usate il comando sudo systemctl mask <servizio>.

Nell'articolo sono state trattate le opzioni più frequentemente utilizzate dell'utility **systemd-analyze**, ma potete sfruttarla anche per molti altri scopi

@italiaNews

VICE

Crediti: https://vice-emu.sourceforge.io

Riscopri il Commodore PET sul tuo PC

Un computer che ha esordito nello stesso anno di "Guerre Stellari" può tornare a ruggire sulla vostra macchina con Linux

Nel lontano 1977 **Commodore** ha lanciato un nuovo computer, poco dopo il suo **KIM-1** (un **microcomputer** che sembrava una grossa calcolatrice). Il **PET** (**Personal Electronic Transactor**), in seguito ridenominato **Commodore CBM** (da **Commodore Business Machines**) per evitare problemi legali in Europa dato che **Philips** distribuiva un computer con quella sigla, era una macchina compatta. Tutto, incluso lo schermo, era all'interno di un grosso **case** di plastica. C'era però un pratico modo per accedere all'hardware, sbloccando la parte frontale dell'unità, che si apriva verso l'alto come il cofano di un'auto. Questa struttura ha avuto un ottimo successo nelle scuole del Nord America e Commodore controllava due terzi del mercato prima del dominio di **Apple**, iniziato negli anni Ottanta. L'emulatore più diffuso per le macchine Commodore si chiama **VICE** (**Versatile Commodore Emulator**) e si può usare per i modelli **C64**, **C128**, **VIC20**, tutti i **PET** e il **CBM-II**. Potete installare VICE dal **gestore di pacchetti**, ma per ottenere l'ultima versione e tutti i file che

Il menu principale di **VICE** è ricco di elementi utili, come **Drive**, in cui potete inserire immagini virtuali di dischi e crearne di nuove per salvare

TIP

Il Commodore PET ha una tastiera particolare e a volte è difficile trovare il tasto giusto. Se vi serve una mano c'è una tastiera virtuale. Premete F12 e scorrete fino all'opzione Virtual Keyboard, poi premete Invio per aprirla. Scegliete quindi un tasto e premete ancora Invio.

Il **look** iconico della gamma è difficile da dimenticare: un computer compatto con monitor integrato prima di **Apple**!

CREDITI: Rama, CC BY-SA 2.0 FR, https://commons.wikimedia.org/wiki/File:Commodore_2001_Series-IMG_0448b.jpg

servono a farla funzionare dovete fare lo scaricamento all'indirizzo **https://bit.ly/3AKVwDh** ed estrarre i contenuti nella vostra **cartella Home**. Questo crea automaticamente una **directory** di nome **vice**. Aprite poi un **terminale** e passate alla cartella nuova. Prima di iniziare l'installazione dovete assicurarvi di avere tutti i pacchetti necessari inserendo il comando che segue e premendo il **tasto Invio**:

```
$ sudo apt update
$ sudo apt-get install build-essential fakeroot dpkg-dev
```

Potreste anche dover installare questi due pacchetti:

```
$ sudo apt install xa65
$ sudo apt install libsdl2-image-dev
```

A questo punto passate alla configurazione, alla compilazione e all'installazione di VICE:

```
$ sudo ./configure
$ sudo make
$ sudo make install
```

Prima di procedere oltre inserite il seguente

comando per lanciare l'emulatore del PET. Viene quindi riprodotta una serie di segnali acustici e appare l'**interprete del BASIC**:

```
$ xpet
```

Se questo non accade, dovete copiare i contenuti della cartella **data/PET** dell'archivio scaricato nella vostra installazione. Fatelo con il comando:

```
$ sudo cp /home/$USER/vice-3.5/data/PET/* /usr/lib/vice/PET
```

Il BASIC del Commodore PET

Il **BASIC** è un ottimo **linguaggio di alto livello** che risale al 1964. Semplificando, è un linguaggio leggibile da esseri umani che utilizza parole inglesi. Il BASIC del Commodore PET è in realtà **6502 BASIC** di **Microsoft** ed è molto simile alla versione del **Commodore 64**. In realtà, le basi del linguaggio sono sostanzialmente identiche in tutte le macchine dell'epoca. Nel caso non lo abbiate mai usato, fate una breve prova **printando** "Ciao mondo!". Aprite **xpet** nel terminale. Il codice è di una sola riga:

```
10 print "Ciao mondo!"
```

Nella successiva scrivete **run** per eseguirlo e vedrete sullo schermo la scritta **Ciao mondo!**. Il BASIC usa per le righe numeri che vanno di dieci in dieci, nel caso sia necessario aggiungere altre righe di codice in seguito. Si possono richiamare e potete saltare fino a esse usando **GOTO**, o utilizzare **GOSUB** per le **subroutine**, simili a **funzioni** di **Python**. Completato il test, potete passare a un progetto più serio, ossia uno strumento per calcolare distanze, velocità e il tempo necessario a percorrere un certo tragitto. La prima riga di codice è la numero 10 ed è un **remark**, che opera analogamente a un **commento** in Python. Qui indica il nome del progetto, come per esempio:

```
10 rem "Calcolatore per viaggi"
```

Dovete poi printare dei messaggi a schermo. Ogni riga è un'istruzione che serve a insegnare all'utente come usare l'applicazione. Potreste unirle tutti in un'unica riga, ma su un Commodore PET originale il risultato sarebbe piuttosto confuso e avrebbe bisogno di andare a capo:

```
20 print "Cosa desideri calcolare?"
30 print "Per la velocità, premi 1."
40 print "Per la distanza, premi 2."
50 print "Per la durata, premi 3."
```

Dopo aver dato le indicazioni necessarie, passate alla riga 60 e date all'utente un **prompt** a cui

```
VICE: PET, press "F12" for the menu.

80 if a=2 then goto 220
90 if a=3 then goto 300
100 print "Sorry I do not understand that option. Goodbye"
110 goto 400
140 print "Speed Calculation"
150 input "Distance travelled in miles?"; d
160 input "How long did it take in minutes?"; t
170 print "Your speed in MPH was..."
180 print d / t * 60
190 goto 400
220 print "Distance Calculation"
230 input "What was your average speed in mph?"; s
240 input "How many minutes did you travel for?"; t
250 print "The distance in in miles that you travelled was..."
260 print s * t / 60
270 goto 400
300 print "Time Calcultion"
310 input "How many miles did you travel?"; d
320 input "What was your average speed in mph?"; s
330 print "The total time that you travelled in hours was..."
340 print d / s * 60
350 goto 400
400 end
ready.
save"travel",8
```

Se non volete perdere il vostro lavoro dovete salvarlo. Dopo aver inserito un disco virtuale, potete farlo nel **drive 8** nello stesso modo che con il **Commodore 64**

rispondere. La **keyword INPUT** prende l'**input** della tastiera e in questo caso lo salva in una variabile di nome **a**. L'input inserito deve essere il numero 1, 2 o 3 e il valore viene salvato nella variabile come numero intero:

```
60 input "Premi 1, 2 o 3 e poi Invio"; a
```

Le righe dalla 70 alla 90 sono un'**istruzione condizionale** che verifica quale numero sia salvato nella variabile **a**. Se è **1** il codice salta alla riga 140, se è **2** va alla 220 e se è **3** passa alla 300:

```
70 if a=1 then goto 140
80 if a=2 then goto 220
90 if a=3 then goto 300
```

Nel caso l'utente non risponda con nessuno dei tre numeri accettati, si attiva la riga 100 che gli comunica che l'istruzione non era chiara e che il codice sarà terminato. La riga 110 salta alla fine del codice (riga 400) e termina l'applicazione.

```
100 print "La tua scelta non è chiara. Chiusura applicazione"
110 goto 400
```

TIP

Nei più recenti Commodore PET c'era un comando che causava un danno fisico al monitor CRT. Non usatelo su una macchina originale! Nel vostro PET virtuale, invece, potete usare POKE 59458,62.

» TROVARE I PET IN TV E NEI FILM

La forma iconica di questa macchina è facile da riconoscere e nel corso degli anni il **PET** è comparso anche in diversi film e serie televisive. Negli anni Ottanta era un computer economico e d'impatto da usare per l'integrazione nei set e molti lo hanno adottato a questo scopo. Potete vederlo in diversi episodi di **Airwolf**, **Supercar** e **Il falco della strada (Street Hawk)** e si trova tipicamente nell'ufficio dei cattivi o in qualche laboratorio di ricerca segreto. A testimoniarne la longevità, ne compaiono due in **Buck Rogers**, ambientato nel 2491. Passando al cinema, ne appare uno in **Star Trek II - L'ira di Khan**, ma non sul ponte dell'**USS Enterprise**; è infatti un pezzo d'antiquariato funzionante negli alloggi dell'**Ammiraglio Kirk**. Il PET compare anche in pellicole del ventunesimo secolo: lo vedete infatti nel film del 2013 **Jobs**, alla fiera **West Coast Computer Faire** del 1977. Potete infine ammirarlo come pezzo da museo nella miniserie del 2020 **Devs**.

TIP

Non volete scaricare un emulatore? Usate il PET online all'indirizzo www.masswerk.at/pet. Potete modificare le impostazioni e caricare applicazioni dalla libreria PRG.

La riga 140 è l'inizio della sezione del codice che calcola la velocità. Prima printa il titolo sullo schermo, poi chiede all'utente di inserire la distanza percorsa in miglia. Il valore viene salvato in una nuova variabile, **d**.

```
140 print "Calcolo velocità"
150 input "Che distanza hai percorso in miglia?"; d
```

La riga successiva, la 160, è un'altra domanda che questa volta chiede quanti minuti è durato il viaggio. Il valore viene quindi salvato in un'altra variabile, **t**.

```
160 input "Quanto tempo hai impiegato in minuti?"; t
```

La riga 170 è l'inizio della risposta e mostra una frase che la spiega, seguita dalla riga 180 che fa il calcolo. La distanza è quindi divisa per il tempo percorso in minuti e poi moltiplicata per 60 per ottenere la velocità (in **MPH**).

```
170 print "La tua velocità in MPH è stata di..."
```

Potete caricare il vostro progetto dal disco virtuale usando il comando **load"programma",8**. Basta poi scrivere **run** per avviare il codice. Se lo avessimo saputo 40 anni fa!

» GIOCARE CON IL PET

La macchina originale aveva solo **4 KB di RAM** e una **CPU 6502** da **1 MHz**. Non offriva molta potenza, ma allora le aspettative erano modeste. Il **Commodore PET** all'inizio aveva una serie di giochi, tra cui semplici **avventure testuali** come **Adventureland**, **Mystery Fun House** e **Pyramid of Doom**. Man mano che gli sviluppatori hanno acquisito familiarità con l'hardware, sono iniziati a uscire videogame con una grafica più elaborata e l'uso dell'audio. **Lunar Lander**, un titolo in cui bisognava far atterrare una navicella sulla Luna, usava i caratteri **PETSCII** per ottenere un buon effetto grafico. **Fire!**, in cui si pilotava un elicottero sopra edifici in fiamme, sfruttava un metodo simile. C'era anche una versione per PET di **Space Invaders**, che in quel periodo era un fenomeno in sala giochi. **Dungeon**, del 1979, era un **dungeon crawler** che creava labirinti generati casualmente. I giocatori avanzavano una casella alla volta, raccogliendo bottino e "combattendo" mostri. Il verbo è tra virgolette perché era il computer a decidere chi vinceva e chi moriva. Fino ai primi anni Ottanta, questi giochi erano considerati di ottima qualità... poi l'arrivo del **Commodore 64** ha cambiato tutto!

```
180 print d / t * 60
```

Dopo aver fatto il calcolo, la riga 190 salta alla 400 che termina l'applicazione.

```
190 goto 400
```

La sezione successiva del codice calcola la distanza e la riga 220 ne mostra il titolo:

```
220 print "Calcolo distanza"
```

Le righe 230 e 240 chiedono la velocità media e la durata del viaggio. Questi valori sono salvati nelle variabili **s** e **t**, rispettivamente:

```
230 input "Qual è stata la velocità media in mph?"; s
240 input "Qual è stata la durata in minuti del viaggio?", t
```

La riga 250 è l'inizio della risposta, che ancora una volta spiega all'utente che cosa va a mostrare, ossia la distanza che è stata percorsa.

```
250 print "La distanza che hai percorso, in miglia, è stata..."
```

La riga 260 printa la distanza sullo schermo. Prende quindi la velocità (**s**) in miglia all'ora, la moltiplica per il tempo in minuti (**t**) e divide la risposta per 60 per visualizzare il valore corretto in miglia:

```
260 print s * t / 60
```

La riga 270 fa saltare il codice alla fine:

```
270 goto 400
```

L'ultima sezione gestisce il calcolo del tempo impiegato e la riga 300 ne printa il titolo a schermo:

```
300 print "Calcolo durata"
```

Le righe 310 e 320 fanno delle domande per scoprire la distanza percorsa e la velocità media. I valori sono salvati, rispettivamente, nelle variabili **d** ed **s**:

```
310 input "Quante miglia hai percorso?"; d
320 input "A quale velocità media in mph?"; s
```

La riga 330 inizia a dire all'utente il risultato ottenuto dall'equazione:

```
330 print "La durata totale in ore del viaggio è stata..."
```

Per calcolare la durata, prima si divide la distanza percorsa per la velocità:

```
340 print d / s
```

La riga 350 manda il codice alla riga 400, che è l'ultima, terminando l'applicazione e uscendo.

```
350 goto 400
400 end
```

Con il codice completato, scrivete `run` per avviarlo e iniziare a fare i vostri calcoli.

Come salvare il vostro lavoro

Una volta che spegnete il computer, il codice viene perso. Per salvarlo dovete inserire un **floppy disk virtuale**. Premete **F12** e, usando i tasti, navigate fino a **Drive**. Premete quindi **Invio**. Nel nuovo menu scorrete in basso fino a **Create new disk image**. Nella schermata successiva il tipo di immagine deve essere **D64**. Scorrete fino ad arrivare a **Create**, premete **Invio** una volta e poi di nuovo per inserire il nome del file. Create un nuovo disco chiamato **viaggi**, con l'etichetta **viaggi**. In **Select Unit**, accettate le impostazioni predefinite con il tasto **Invio**. Premete **ESC** per tornare al codice e poi di nuovo **F12**, per rientrare in **Drive**. Selezionate **Attach disk image to drive 8** nel menu, premete **Invio** e poi di nuovo **Invio**. Inserite il nome del file

del disco (**viaggi**) e premete di nuovo **Invio**. Ora che il disco è inserito, premete **ESC** per tornare all'interprete del BASIC. Per salvare il codice come un programma di nome **viaggi** scrivete quanto segue e premete il tasto **Invio**:

```
save"viaggi.prg",8
```

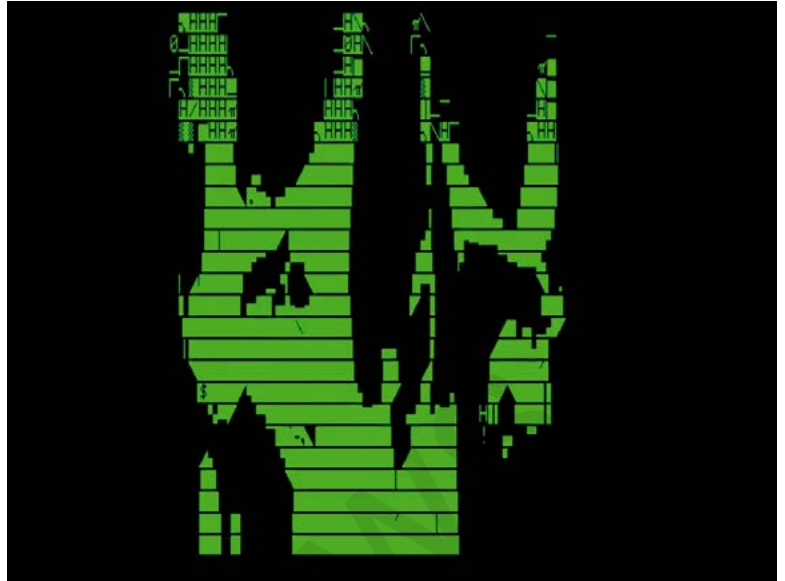
Il codice è ora salvato in un'immagine disco virtuale sulla vostra macchina con **Linux**. Potete caricarlo in qualsiasi momento premendo **F12**, andando in **Drive** e selezionando **Attach disk image to drive 8**.

Giocare con i videogame

Il Commodore PET non era in grado di gestire gli **sprite** e aveva delle capacità audio modeste quindi non era esattamente la macchina da gioco ideale, ma questo non ha impedito ad aziende e programmatori di produrre **videogame** per tutti i modelli di questo computer. **Dog Star Adventure** è una sorta di incrocio tra **Star Wars** e **Star Trek** che non ha una grafica spettacolare ma è un modo piacevole per passare un pomeriggio. **Lunar Lander** vi fa provare un atterraggio lunare ed è un solido simulatore per il periodo. Tutti i computer hanno i loro giochi di carte e con **CBM Draw Poker** potete sfidare il PET. Online si trovano diversi titoli ma, malgrado l'età avanzata, sono in genere ancora protetti da **copyright** quindi, a meno che non possediate l'originale, non potete scaricarli. Trovate i videogame sotto forma di disco o di cassetta e potete fare il **download** dal terminale. Navigate fino alla cartella in cui tenete i giochi e inserite nel terminale il seguente comando:

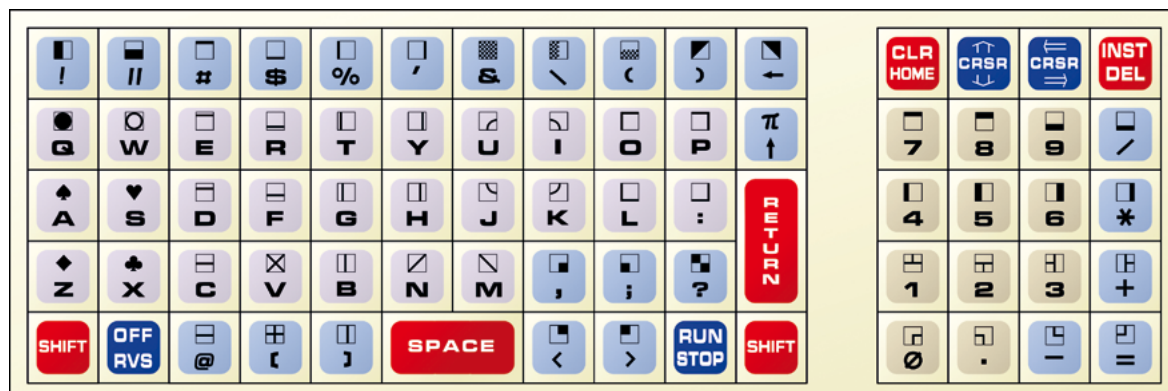
```
xpet <<gioco>>
```

La demoscene e i fan di oggi

La serie Commodore PET non è mai spiccata come un potente strumento per la creazione multimediale. Il Commodore 64, uscito pochi anni dopo, ha realmente lanciato la **demoscene**. Recentemente però sono state pubblicate diverse **demo** per il PET e ci ha colpito **Bionary** di **oobc**. Pubblicata in occasione del **demo party Zoo 2017**, è un'esplosione di bella grafica con una linea di basso ottima per le capacità della macchina. La potete vedere all'indirizzo **https://youtu.be/PBykK8oNyX4**. Se la volete per la vostra emulazione del PET, potete trovare il file su **www.pouet.net/prod.php?which=71928**. Scaricate

Bionary di **oobc** è un capolavoro sia per l'uso dei **caratteri PETSCII** nella creazione di elaborate animazioni, sia per l'audio. Le animazioni del teschio sono degne di **Cyberpunk 2077**!

ed estraete il **file ZIP**, poi aprite la demo usando il comando **xpet** dal terminale:

```
$ xpet bionary.prg
```

Questo computer del 1977 ha ancora un buon numero di fan nel 2021 e potete comprare vari accessori correlati. Ci sono i soliti sostituti con **schede SD** per i drive dei dischetti, tra cui il più noto è lo **SD2PET** (lo trovate **online** su **www.thefuturewas8bit.com**), e anche una scheda **RAM/ROM** per sostituire i **chip** che non funzionano più. Potete persino costruire il vostro PET "fai da te" e l'azienda inglese **Tynemouth Software** ha prodotto un **kit** per un mini Commodore PET (**MINIPET 40/80**) con anche la tastiera. A 299 £ (circa 352 €) non è economico, ma è un progetto divertente che permette di imparare molto. Il PET non ha avuto la popolarità del suo successore, il Commodore 64, ma nel 1977 il mondo era ancora all'inizio dell'era degli **HOME COMPUTER** e questo modello, insieme al **TRS-80** e all'**APPLE II**, era tra i migliori per la sua fascia di prezzo. Anche se non offriva il massimo per grafica, audio o selezione di giochi, in alcuni Paesi è stato il simbolo di un'era ed era sicuramente una macchina pratica e facile da usare che ha fatto la storia. LXP

La tastiera del **PET 2001** era molto futuristica. È un misto di lettere, numeri e segni di interpunzione, con caratteri e icone nascosti per molti dei tasti

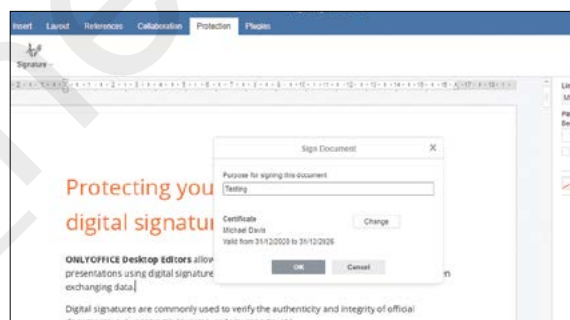
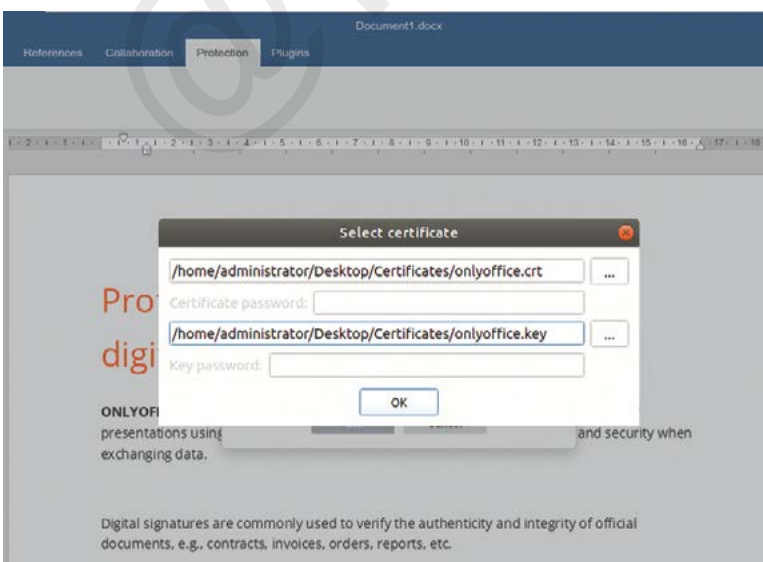
ONLYOFFICE

Crediti: www.onlyoffice.com

Proteggete i vostri documenti

Firme digitali, password e filigranature vi permettono di aggiungere un nuovo livello di sicurezza ai vostri file più preziosi

Oggi la sicurezza dei documenti è sempre più importante. Le firme digitali vengono comunemente usate per verificare l'autenticità e l'integrità di documenti ufficiali come contratti, ordini e fatture. Dimostrano che il file in questione è stato inviato da voi e che i suoi contenuti non sono stati modificati, dato che il documento non si può più **editare** dopo l'apposizione della firma elettronica, che viene invalidata in caso contrario. Per poter apporre una firma digitale, vi serve un **certificato digitale** di una **Certificate Authority**. Vi fornisce infatti le chiavi usate per validare l'autenticità delle firme digitali. Per creare la firma viene usata una **chiave privata** a cui può accedere solo il proprietario del certificato. Chi riceve il documento usa quindi una **chiave pubblica** che conferma l'autenticità della firma digitale del mittente. Per aggiungere una firma digitale invisibile, create un nuovo file o apritene uno esistente con **OnlyOffice Desktop**

Selezionare un certificato digitale in **OnlyOffice**

In **Purpose for signing this document** mettete lo scopo della firma

Editors. Passate alla scheda **Protection**, fate click sulla freccia accanto al pulsante **Signature** e poi selezionate l'opzione **Add digital signature**. Nella finestra che appare inserite lo scopo della firma, poi selezionate un certificato digitale caricando i file **.crt** e **.key** prima di fare click su **OK** nella finestra **Sign Document**. Il vostro documento è quindi firmato e protetto da modifiche. Come anticipato, questo tipo di firma non si vede nel testo, ma nella barra degli strumenti a destra potete trovarne impostazioni e dettagli, incluso il nome del proprietario del certificato, data e ora della firma, oltre a opzioni per rimuoverla o aggiungerne un'altra. Se decidete di modificare un documento firmato, compare un messaggio che vi dice che è protetto. Fate click su **Edit anyway** per rimuovere la firma.

Aggiungere una riga per la firma

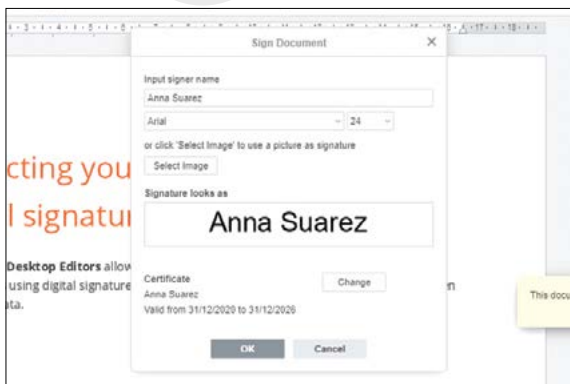
In questo caso si vede una riga per la firma nel documento, in cui altri utenti possono mettere la propria. Per mettere la riga, create un documento o apritene uno esistente con OnlyOffice Desktop Editors e mettete il cursore del mouse nel punto in cui volete apporla. Passate alla scheda **Protection**, fate click sulla freccia accanto al pulsante **Signature** e poi selezionate l'opzione

@Italia_news

Impostare una password protegge i documenti in locale

Add signature line. Nella finestra che compare, compilate i campi del nome, del titolo del firmatario, dell'**e-mail** e delle istruzioni per chi deve firmare. Se serve, spuntate l'opzione **Show sign date in signature line** per mostrare la data. Fate click su **OK** e salvate il documento. Compare quindi la riga e potete firmare voi stessi, oltre che richiedere ad altri di farlo. Si possono anche aggiungere più righe se ci sono molti firmatari, ma dovete farlo prima di firmare voi il documento.

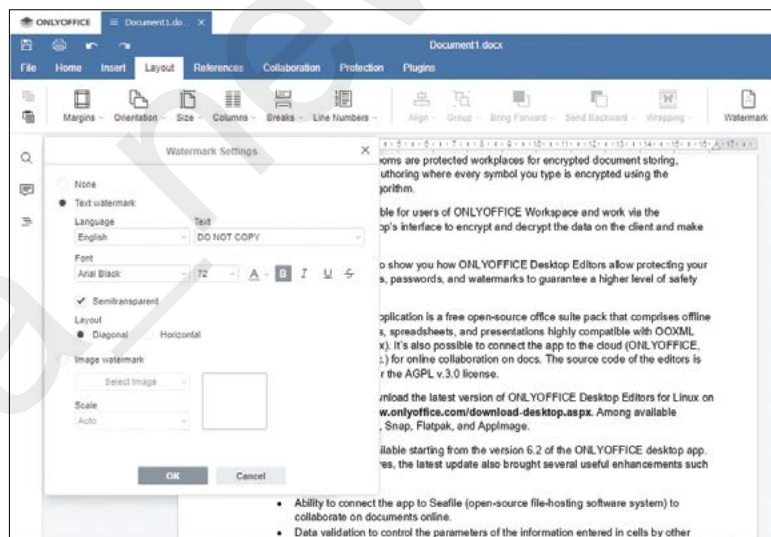
Richieste di firma digitale

Se ricevete un documento che richiede la vostra firma, vedete una **finestra pop-up** che vi informa di questo fatto. Per firmare, fate doppio click sulla riga della firma e selezionate l'opzione **Sign** all'interno del menu che si apre. Inserite quindi il vostro nome nel campo **Input signer name** e formattate il testo scegliendo il tipo di carattere da utilizzare e le sue dimensioni. In alternativa, potete fare click sul pulsante **Select image** e caricare un'immagine della firma. Verificate l'anteprima, scegliete un certificato e fate click su **OK** nella finestra **Sign Document**. Le firme risultano quindi visibili a tutti e il documento è protetto dalla modifica. Nell'applicazione per desktop di **OnlyOffice**, potete proteggere con una password i file salvati localmente. Per farlo, create o aprite un file con **OnlyOffice Desktop Editors**, entrate nella scheda **Protection**, selezionate **Encrypt** e inserite una password solida. Se volete sbloccare il documento o cambiare la password, selezionate **Remove password** nella stessa pagina in cui la avete impostata in precedenza.

Potete aggiungere a un documento più righe per le firme

Aggiungere le filigranature

Le filigranature, chiamate anche con il nome inglese **watermark**, aiutano a ridurre il rischio di ridistribuzione non autorizzata del vostro lavoro. Si tratta di un testo o di un'immagine posizionati sotto il livello del testo principale. Per aggiungere la filigranatura, create o aprite un documento con OnlyOffice Desktop Editors, passate alla scheda **Layout**, fate click sull'icona **Watermark** e selezionate **Custom Watermark**. Regolate quindi i parametri disponibili nella finestra che si apre. Se aggiungete una filigranatura testuale, potete selezionare la lingua, il testo del watermark, il nome e le dimensioni del carattere, lo stile e la disposizione. Si può anche applicare l'opzione per la semitrasparenza. Se invece la filigranatura è un'immagine, potete scegliere il file sorgente e la scala. Al termine dell'operazione, fate click su **OK** per aggiungerla al vostro documento. LXP

TIP

OnlyOffice è disponibile nei formati DEB, RPM, Snap, Flatpak e AppImage. È preinstallato in Linuxfx, SparkyLinux, Linspire, Escuelas Linux e Linkat e disponibile negli store di Mint e Zorin OS.

Aggiungere una filigranatura a un documento ne scoraggia la distribuzione non autorizzata

» PRIVACY E SICUREZZA

Le **Private Room** di **OnlyOffice** sono spazi di lavoro protetti per salvare, condividere e modificare documenti. Ogni simbolo che scrivete viene **criptato** con l'**algoritmo AES-256**, che risulta sostanzialmente inviolabile. Le Private Room sono disponibili per gli utenti di **OnlyOffice Workspace** e usano l'interfaccia desktop di OnlyOffice per **criptare** e **decriptare** i dati sul **client** e stabilire un **endpoint** di sicurezza. **OnlyOffice Desktop Editors**, come approfondito nel corpo dell'articolo, permette di proteggere i file con firme digitali, password e filigranature. L'applicazione per desktop di OnlyOffice è una **suite** per la produttività personale Open Source che include **editor offline** per documenti di testo, fogli di calcolo e presentazioni altamente compatibili con i formati **Office Open XML** (**OOXML**) come **.docx**, **.xlsx** e **.pptx**. Potete anche collegare il programma al **cloud** (**OnlyOffice**, **Nextcloud**, **ownCloud**, **Seafile**) per collaborare **online** sui documenti. Il codice sorgente per gli editor è disponibile su **GitHub** con licenza **AGPL v.3.0**.

DESKREEN

Crediti: www.deskreen.com

Usa il cellulare come secondo display

Ecco un modo semplice per poter visualizzare i contenuti del monitor del vostro PC sul browser del vostro cellulare, tablet o laptop

Non sempre c'è lo spazio sulla scrivania (o il budget) per aggiungere un secondo monitor, ma averlo può rivoluzionare il modo in cui usate il PC. **Deskreen** risolve il problema permettendovi di convertire altri dispositivi, come cellulari, **tablet**, **laptop** e persino **smart TV**, in un **display** aggiuntivo per il vostro computer. Può riprodurre tutta la scrivania consentendovi

Deskreen vi permette di monitorare una singola applicazione o l'intera scrivania del vostro computer su un dispositivo mobile

di monitorare il computer in remoto oppure potete usarlo per vedere i contenuti di uno specifico programma. Con le giuste conoscenze (vedi box in questa pagina) potete persino fargli estendere il vostro desktop. Deskreen supporta sia **X** sia **Wayland** e trasmette i contenuti del vostro display ai dispositivi mobili sulla vostra rete locale, con o senza fili, usando il **protocollo WebRTC**. La maggior parte del carico di lavoro ricade sul PC quindi potete usare anche cellulari e tablet meno recenti, purché abbiano un browser che supporta WebRTC. In **Android**, basta avere **Chrome 28**, **Firefox 22** o le versioni successive. Se volete sfruttare un vecchio **iPhone** o **iPad**, deve avere almeno **iOS 11**. Potete scaricare Deskreen come file **.deb** or **AppImage** (che non richiede installazione) da **www.deskreen.com**. Dopo il **download** fate doppio click sul file .deb per installarlo con **Ubuntu Software**, o fate click con il tasto destro sul file AppImage, scegliete **Properties > Permissions** e spuntate **Allow executing file as a program**. Al primo avvio, compare

» ESTENDERE IL VOSTRO DISPLAY CON DESKREEN

Come impostazione predefinita, quando condividete l'intero schermo **Deskreen** lo specchia, il che è utile se volete tener d'occhio in remoto dei compiti in esecuzione. Se la scheda grafica del vostro PC ha più porte, potete spingervi oltre collegando un **adattatore Virtual Display** in un **socket HDMI** o **DisplayPort** libero. Quando lo fate, il computer pensa di avere a tutti gli effetti un secondo monitor e potete vederlo sul vostro dispositivo mobile con Deskreen. Aggiungere un secondo display in questo modo vi permette di estendere il vostro desktop, consentendovi di trascinare finestre dal monitor del computer allo schermo del dispositivo mobile, dove potete vederlo e controllarlo direttamente attraverso il mouse e la tastiera. L'utilizzo di un adattatore è il metodo più semplice per estendere il desktop, ma se non avete problemi a lavorare con uno **script in Bash**, potete provare **Virtual Display Linux** (**https://github.com/dianariyanto/virtual-display-linux**). Gli utenti di schede grafiche **Nvidia** devono però adottare un approccio diverso: seguite il collegamento in fondo al file **readme** per trovare le indicazioni per utilizzare l'opzione **TwinView** di questo produttore.

Gli sviluppatori di **Deskreen** sono al lavoro per supportare **Virtual Display** senza adattatore

@italia_news

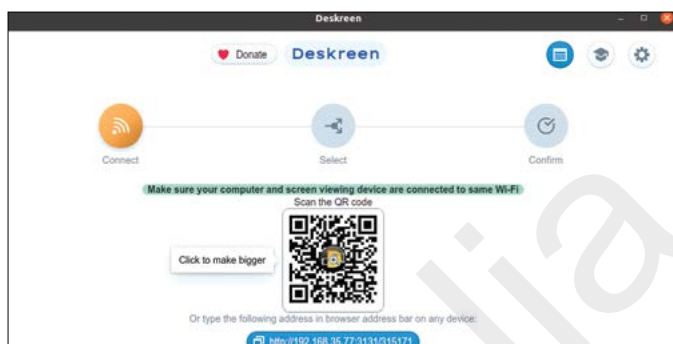
la finestra principale del programma e dovete confermare la lingua e scegliere il tema. Nel box in questa pagina potete vedere come fare la prima connessione al browser remoto.

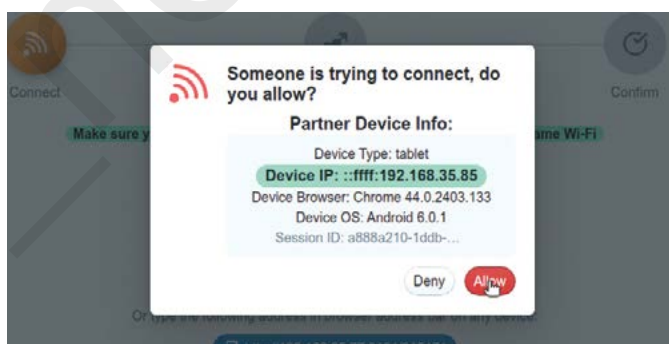
Controllare la connessione

Il browser del secondo display mostra la finestra dell'applicazione o la scrivania. Se lo usate per monitorare un singolo programma che è nascosto sul vostro desktop potete controllarlo solo portandolo in primo piano nello schermo principale con il mouse o la tastiera, mentre il desktop viene fedelmente riprodotto. In alto trovate il logo di Deskreen e alcuni controlli. Potete sospendere la connessione con il pulsante di pausa, fare click su quello delle impostazioni per ruotare lo schermo o regolare la qualità del display se la connessione di rete o l'hardware fanno fatica. C'è anche un'opzione a schermo intero. Anche se non potete interagire sul cellulare con gli elementi visualizzati potete zoomare con le dita per facilitare la lettura. Trovate infine l'opzione **Default Video Player,** che funziona solo con il browser predefinito del dispositivo (**Chrome** in Android o **Safari** in iOS) e vi serve solo se volete ruotare lo schermo. Uno dei punti di forza di Deskreen è che potete usarlo con più display remoti. Dopo aver impostato la prima connessione, fate click su **Connect New Device** per crearne altre. Naturalmente questo comporta un maggior carico per il vostro PC e siete anche limitati dalla velocità della vostra connessione di rete. Al termine della procedura usate il pulsante **Connected Devices** per vedere quali finestre sono condivise con quali dispositivi e per terminare una o più connessioni. L'uso più popolare di Deskreen su cellulare o tablet è come display dedicato per **utility** specifiche che volete seguire ma che non richiedono interazioni. Potete per esempio tener d'occhio l'utilizzo delle risorse in **System Monitor** o seguire i progressi di una sezione di codifica in **Handbrake** mentre lavorate ad altro sul computer. LXP

TIP

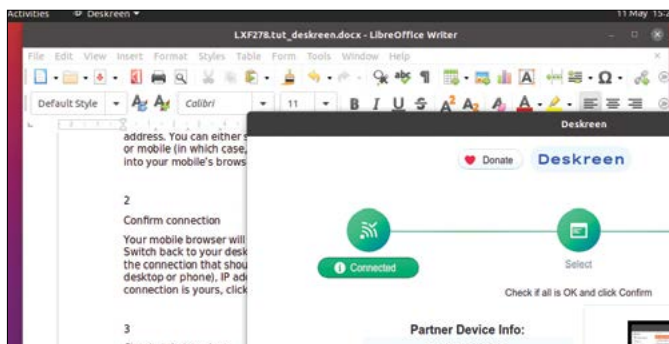
Visitate la pagina Github del progetto (https://github.com/pavlobu/deskreen) per avere informazioni sulle ultime release e collegamenti ai forum degli utenti.

COLLEGATE IL TABLET O IL CELLULARE A DESKREEN

1 Scansite il codice o inserite un URL

Aprendo **Deskreen** vedete la pagina **Connect** che mostra un **Codice QR** e un indirizzo Internet. Potete scansire il **QR code** con un'app dedicata sul vostro **tablet** o cellulare (nel qual caso vi rimanda a un **link** su cui fare click) oppure copiare direttamente il collegamento nel **browser** del vostro dispositivo mobile.

2 Confermate la connessione

Il browser del dispositivo mobile vi comunicherà che dovete fare click su **Allow in Deskreen**. Tornate al desktop, che mostra ora informazioni sulla connessione per il vostro dispositivo, indicandone tipo (tablet, desktop o cellulare), **indirizzo IP**, browser e sistema operativo. Dopo aver verificato i dati, fate click su **Allow**.

3 Scegliete cosa condividere

Potete scegliere se condividere l'intero schermo o la finestra di una singola applicazione. Nel primo caso potete specchiare il monitor (per estendere il vostro **display** fate riferimento al box nella pagina accanto) mentre nel secondo dovete scegliere la finestra di un programma usando le miniature come guide.

4 Confermate ed è tutto pronto!

Visionate il secondo dispositivo e lo schermo o l'app condivisi. Fate click su **Confirm** se va tutto bene o su **No, I need to share other things** se volete fare modifiche. Dopo aver confermato, vedrete un messaggio relativo. Passate al dispositivo mobile dove apparirà la vostra condivisione.

Crediti: https://www.ubuntu.com

Crea un sistema NAS completo con Linux

Un'abbondante porzione di Ubuntu Server, un pizzico di Cockpit e una spolverata di Docker sono la ricetta perfetta per un server NAS

Prima di tutto dovete scegliere un sistema operativo adatto per il vostro **server NAS** e, dato che sarà **headless**, un'ottima soluzione è **Ubuntu Server 20.04 LTS**, che non ha componenti come il **desktop**, in questo caso inutile dato che tutto viene fatto da riga di comando. Dopo la configurazione, però, installerete un **add-on** (**Cockpit**) che permette di amministrare facilmente il server da un **browser** sulla vostra rete locale. È stata scelta la versione LTS di Ubuntu Server per ridurre il numero di aggiornamenti al sistema operativo da introdurre durante la vita del server.

Iniziate il processo di installazione

Andate alla pagina **www.ubuntu.com/download**, fate click su **Ubuntu Server** e poi su **Option 2 – Manual server installation**. Troverete come opzione principale l'ultima versione LTS. Lo scaricamento è molto più ridotto di quello dell'edizione completa per desktop ed è di poco più di 1 GB. Dopo il **download**, potete copiarlo su un **DVD** o su una **scheda USB** vuota da almeno 2 GB. Una volta creato il supporto di installazione, collegatelo o inseritelo nel vostro PC che userete come server prima di accenderlo. Se il **BIOS** non è configurato per avviarsi da drive USB, premete il tasto **F11** (o **F12**) all'avvio per far apparire il **menu di boot**, poi selezionate il vostro supporto USB per iniziare l'installazione. L'**installer** di Ubuntu Server, a differenza di quello per la versione desktop, è testuale. Quando compare il menu di **GRUB**, lasciate selezionato **Install Ubuntu Server** e premete il tasto **Invio**. Sono visualizzati dei messaggi durante il test del drive e l'avanzamento dell'installazione. Alla fine compare un menu testuale per la configurazione.

Il **wizard** per l'installazione di **Ubuntu Server** è abbastanza semplice da usare

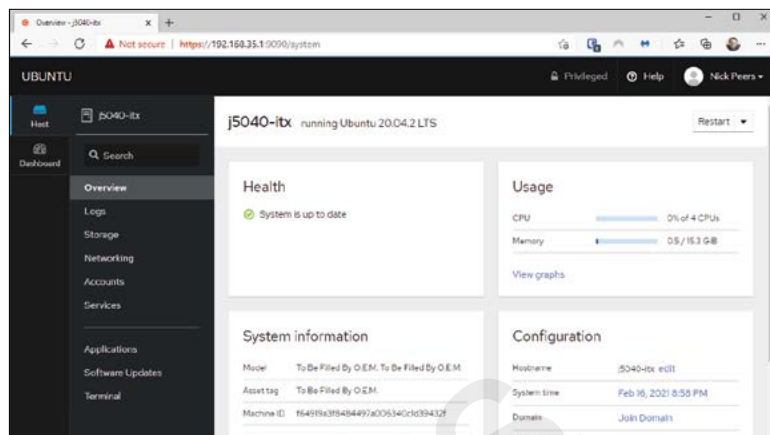
```
Edit ens33 IPv4 configuration
IPv4 Method:    [ Manual         ▾ ]

      Subnet:   192.168.35.0/24
     Address:   192.168.35.1
     Gateway:   192.168.35.254
Name servers:   1.1.1.1,1.0.0.1
                IP addresses, comma separated
Search domains:
                Domains, comma separated
                      [ Save    ]
                      [ Cancel  ]
```

Configurate il vostro sistema

Scegliete la lingua, premete **Invio**, poi selezionate il tipo di tastiera. Il passo successivo è il più critico, perché riguarda la configurazione della rete. I server devono avere **indirizzi IP statici**, che permettono di trovarli facilmente in rete. Per prima cosa premete due volte il **tasto freccia in alto** per

@italianews

scegliere l'interfaccia attuale (**ETH** o **WLAN**) e premete **Invio**. Usate il **tasto freccia in basso** per selezionare **Edit IPv4**, premete **Invio**, poi premete **Invio** ancora per passare dall'impostazione **Automatic** (**DHCP**) a **Manual**. Vi viene quindi chiesto di selezionare le impostazioni: usate i tasti freccia per spostarvi tra i campi. Normalmente, sotto **Subnet** dovreste inserire **255.255.255.0**, ma scrivete invece `192.168.x.0/24`, sostituendo `x` con il numero mostrato sotto **DHCPv4.** Inserite l'indirizzo IP desiderato nel campo **Address**, assicurandovi che non vada in conflitto con altri dispositivi di rete, e quello del vostro **router** nel campo **Gateway**. In **Name servers** inserite i vostri **server DNS**, tipicamente `8.8.8.8,8.8.4.4` per **Google** o `1.1.1.1,1.0.0.1` per **Cloudflare**. Il campo **Search domains** è richiesto solo se volete aggiungere un dominio locale agli **hostname** nella vostra rete, per esempio **mypc.ilmiodominio.com**. Se non siete sicuri, lasciatelo vuoto. Al termine della procedura, selezionate **Save** e premete **Invio**. Una volta che i cambiamenti sono stati applicati, selezionate **Done** e premete **Invio**. Saltate il campo **Proxy** e lasciate selezionati gli indirizzi dei **mirror** di Ubuntu predefiniti. Passate quindi a configurare lo spazio di archiviazione. Assumendo che l'intero disco di avvio del vostro sistema sia dedicato a Ubuntu, lasciate selezionato **Use an entire disk** e **Set up this disk as an LVM group** quindi proseguite, accettando il riepilogo del **filesystem** (gli utenti avanzati possono selezionare le impostazioni personalizzate per creare partizioni separate per le cartelle **root** e **home** se lo desiderano). Il punto di non ritorno è la finestra di dialogo **Confirm destructive**. Selezionate **Continue** per accettare i cambiamenti non modificabili apportati. Vi viene quindi chiesto di impostare il vostro **account utente** e il nome del computer che lo identificherà sulla rete. Potete poi scegliere se installare **OpenSSH** per avere un accesso remoto sicuro al server attraverso strumenti da riga di comando come **Putty**. Non è però necessario in questo caso, perché userete **Cockpit**.

La vista principale di **Cockpit** offre una pratica panoramica del sistema

Installare componenti aggiuntivi

Si può quindi installare una serie di componenti aggiuntivi da pacchetti **snap**. Consigliamo sicuramente **Docker** e **Canonical-livepatch**. Fate poi click su **Done**. Compare una schermata di **log** che mostra l'avanzamento del processo di installazione. Attendete che siano applicate le **patch per la sicurezza** poi selezionate **Reboot Now** quando vi viene proposto come opzione. Togliete il supporto di installazione dal PC e fate avviare Ubuntu Server. Fate il **login** e configurate Cockpit. Per installarlo usate i comandi:

```
$ sudo apt update && sudo apt upgrade
$ sudo apt install cockpit -y
$ sudo systemctl start cockpit
```

Verificate che sia attivo e in esecuzione con il comando (`sudo systemctl status cockpit`). Potete ora amministrare il server da qualsiasi computer sulla vostra rete locale.

Iniziare a usare Cockpit

Per accedere remotamente al vostro server, andate all'indirizzo **https://192.168.x.y:9090/** sostituendo a **192.168.x.y** l'indirizzo del vostro server. Non preoccupatevi degli avvisi di sicurezza del browser: derivano dal fatto che Cockpit genera un **certificato SSL autofirmato** per l'**indirizzo IP**.

» NON DIMENTICATEVI I BACKUP

Ogni tanto vi capiterà di provare sul vostro server qualcosa che non funziona o che danneggia qualcos'altro nel processo. Grazie a **Timeshift** potete tornare all'ultima versione salvata del sistema e rimediare a qualsiasi problema. Installatelo con:

```
$ sudo add-apt-repository ppa:teejee2008/timeshift
$ sudo apt update && sudo apt install timeshift
```

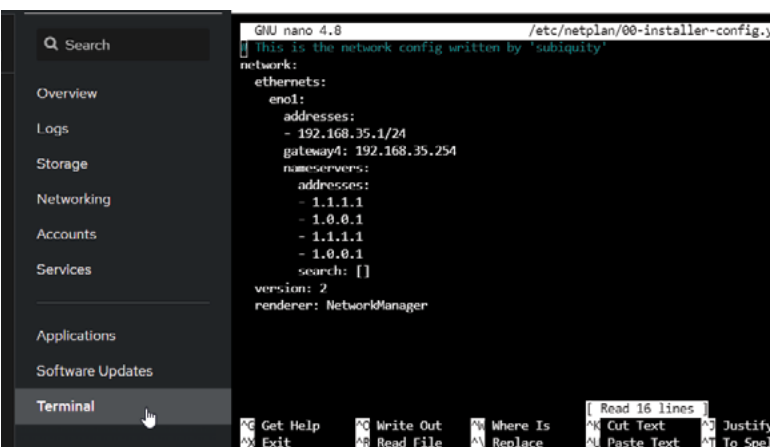
Scrivete `timeshift` dopo l'installazione per vedere un elenco dei comandi disponibili. Per eseguire il vostro primo **backup**, salvato nella vostra partizione di avvio, usate il comando:

```
$ sudo timeshift --create --comments "Backup 01"
```

Programmare i backup da riga di comando non è semplice e dovete saper usare **crontab**, ma potete farli manualmente in momenti critici, per esempio prima di un aggiornamento importante o dell'installazione di nuovo software. Nel caso doveste ripristinare un backup, utilizzate il comando:

```
$ sudo timeshift --restore
```

Vi verrà chiesto di scegliere a quale versione del sistema volete tornare (è importante saperle identificare tramite i commenti) e se volete reinstallare il **bootloader GRUB**. Dopo aver riavviato vi troverete nella situazione che avevate prima che si verificassero i problemi. Se preferite gestire i backup dal **browser**, potete dare un'occhiata a **BorgBackup** (**https://borgbackup.readthedocs.io**). La configurazione di questo programma risulta molto più laboriosa di quella di **Timeshift**, ma il risultato finale può valerne la pena.

Quando appare l'avviso, create un'eccezione per permettere l'accesso e andare alla schermata di login. Inserite nome utente e password che avete impostato durante l'installazione per l'accesso a Ubuntu Server. Assumendo che siate su un PC sicuro, lasciate selezionato **Reuse my password for privileged tasks** e fate click su **Log In**. Vi trovate quindi nella schermata principale di Cockpit, nella sezione **Overview** che offre una panoramica dello stato del vostro server oltre che una scorciatoia per chiudere e riavviare il sistema. Se ricevete la notifica che non avete i diritti per riavviare il server, significa che non avete lasciato selezionato **Reuse my password for privileged tasks** al login. Risulteranno disabilitate anche altre funzioni, come il controllo degli aggiornamenti, e nei **log di sistema** vedrete un messaggio di errore che potete ignorare.

Aggiungete questa riga per assicurarvi che **Cockpit** funzioni bene con **Ubuntu 20.04**

Risolvere i problemi in rete

Sulla sinistra della schermata trovate le scorciatoie per tutte le funzioni offerte da Cockpit. Prima di procedere, però, dovete sistemare un **file di configurazione** di Ubuntu 20.4 per risolvere un problema del sistema operativo con gli aggiornamenti del software e di rete e poi riavviare. Per farlo, selezionate la scorciatoia **Terminal**, che funziona come in Ubuntu. C'è il completamento automatico per velocizzare l'inserimento dei percorsi e dei nomi dei file ed è pienamente compatibile con il **blocco note**, il che consente di copiare e incollare facilmente comandi da altre fonti, incluse pagine Web. Nel terminale, scrivete:

```
$ sudo nano /etc/netplan/00-installer-config.yaml
```

Si apre un file di configurazione. Aggiungete la riga che segue sotto `version: 2` :

```
renderer: NetworkManager
```

Premete i tasti **Ctrl** + **X** seguiti da **Y** e **Invio** per salvare le modifiche. Riavviate il server e rifate il login. L'opzione **Dashboard** si collega a dei grafici per le prestazioni (accessibili anche altrove e rimossi da qui nelle **build** più recenti di Cockpit), mentre **Host** offre le scorciatoie per tutte le parti del sistema che potete controllare con il programma.

Configurare lo spazio di archiviazione

Ora che avete Ubuntu Server e Cockpit funzionanti nel drive di avvio, dovete configurare anche gli altri supporti fisicamente installati. Prima di farlo, però, organizzate un valido sistema di backup (vedi box **Non dimenticatevi i backup**). La configurazione dei drive è un processo abbastanza semplice, ma presuppone che sappiate come **Linux** vede dischi e partizioni. Sono sostanzialmente montati come cartelle all'interno del vostro **filesystem**, quindi dovete impostare dei **punti di montaggio**. Aprite il terminale e inserite:

```
$ mkdir ~/media
$ mkdir ~/media/bay1
```

Ora potete **partizionare**, **formattare** e **montare** il vostro primo drive. Passate alla sezione **Storage** e scorrete fino a quella **Drives**, che elenca tutti i supporti fisici installati, che qui si assume non siano formattati. **/dev/sda** si riferisce al disco di avvio, che avete configurato durante l'installazione, quindi potete passare a **/dev/sdb**. Fateci click sopra per accedere alla sua schermata delle proprietà in cui potete vedere che è elencato come **unrecognised data** sotto **Content**. Per risolvere il problema, fate click su **Create Partition Table**, scegliete il tipo di partizione (**MBR** o **GPT**, lasciate quest'ultimo selezionato se non siete sicuri) e poi fate click su **Format**. Con la tabella di base sistemata, fate click su **Create Partition** per aprire una nuova finestra di dialogo che rende facile partizionare il drive, usando se necessario lo **slider** per creare più partizioni una dopo l'altra.

» AGGIORNATI CON GLI SNAP

Tradizionalmente, il software in **Ubuntu** viene installato attraverso archivi di pacchetti detti **repository**, che rimangono una fonte vitale di programmi per il vostro server. Ci sono tre repository ufficiali per **Ubuntu 20.04 LTS**: **focal**, **focal-updates** e **focal-backports**. Potete vedere i loro contenuti all'indirizzo **https://packages.ubuntu.com** e installare le applicazioni da riga di comando quando entrate in **Ubuntu Server** per la prima volta. Il limite di questi archivi è che molti dei programmi sono aggiornati al momento della pubblicazione di Ubuntu 20.04 e gli **update** degli sviluppatori saranno ignorati fino al prossimo **major upgrade** (la versione **22.04** dovrebbe uscire a maggio del 2022). Un esempio del problema è **Cockpit**, che è fermo alla versione **215** in Ubuntu 20.04 malgrado siano state pubblicate nel frattempo decine di aggiornamenti. Molte applicazioni offrono i propri repository che funzionano come quelli di Ubuntu: dopo l'installazione potete accedervi con il comando `apt` e venite informati delle nuove uscite. Lo sviluppatore di Ubuntu, **Canonical**, ha inoltre creato i **pacchetti snap**, che si installano con il comando `snap` e si aggiornano automaticamente appena il produttore rilascia una nuova versione. Un programma può essere presente in vari repository e anche come pacchetto snap (usate il comando `snap find appname` per verificarlo) e installarlo in questo modo invece che dai repository vi assicura che rimanga sempre aggiornato. Se l'applicazione che volete non è disponibile come snap, visitate il sito del produttore per vedere se ha un suo repository. Nel caso aggiungetelo e usate `apt` per installare il pacchetto desiderato.

È consigliabile selezionare come tipo di file **ext4** invece dell'opzione raccomandata **xfs**, che è più pensata per sistemi operativi per grandi organizzazioni, come **Red Hat**. Date un nome al vostro drive, per esempio in base al punto di montaggio (qui **Bay1**), poi inserite il percorso del **mountpoint**: **/home/<nomeutente>/media/bay1**. Lasciate selezionato **Mount now** e fate click su **Create Partition**. Al termine del processo, potreste vedere un avviso accanto al drive. Nel caso fateci click sopra e scegliete di montare subito, il che completa il lavoro e vi dà accesso al disco e ai suoi contenuti attraverso il punto di montaggio selezionato. Se avete altri drive installati, ripetete la procedura per ciascuno di essi.

Condividere le cartelle

Potete quindi passare a configurare le vostre cartelle condivise. Per assicurarvi di potervi accedere da qualsiasi sistema operativo dovete installare **Samba**. Lo fate con i seguenti comandi, che creano anche una cartella condivisa nel vostro supporto appena formattato:

```
$ sudo apt install samba
$ mkdir ~/media/bay1/shared
$ sudo nano /etc/samba/smb.conf
```

Aggiungete quindi il blocco di codice che segue in fondo al file (senza dimenticarvi di sostituire a `<nomeutente>` il vostro **username**):

```
[shared]
    comment = Ubuntu Shared
    path = /home/<nomeutente>/media/bay1/shared
    read only = no
    browsable = yes
```

Salvate il file, uscite da Samba e scrivete:

```
$ sudo service smbd restart
$ sudo ufw allow samba
```

Assegnate infine una password speciale per Samba (che deve naturalmente essere diversa da quella del sistema) al vostro **account**:

```
$ sudo smbpasswd -a <nomeutente>
```

Ora potete accedere alle cartelle condivise da altri dispositivi sulla vostra rete usando il nome utente e la password appena configurati. Se non vengono visualizzati automaticamente, scrivete `\\192.168.x.y` (sostituendo `192.168.x.y` con l'indirizzo IP del vostro server) in una finestra di **Folder Explorer** per poter accedere manualmente alle vostre condivisioni.

Inizia la parte divertente

Ora che avete impostato le basi, iniziate a installare i pacchetti che vi servono. Se volete trasferire dati (come raccolte di file multimediali) in modo facile e veloce da altri dispositivi sulla vostra rete senza dover aprire cartelle condivise, potete usare uno strumento di sincronizzazione come **Syncthing** (**https://syncthing.net/downloads/**). Se vi serve configurare un **media server**, sia **Plex** sia **Jellyfin** sono delle ottime scelte e vanno installati direttamente nel server dai rispettivi repository per poter sfruttare appieno la **transcodifica hardware**.

Per ulteriori suggerimenti per ottenere software per il vostro server fate riferimento al box **Aggiornati con gli snap** nella pagina accanto.

Lavorare sicuri con Docker

È meglio eseguire alcuni servizi nei propri ambienti isolati per motivi di sicurezza e stabilità. In questo modo, infatti, se il servizio incontra un problema, non può causare danni al resto del sistema.
Un modo per ottenere questo risultato potrebbe essere creare diverse **macchine virtuali**, ma impostare una **VM** con il suo sistema operativo non è pratico e vi fa sprecare le preziose e limitate risorse del vostro server. Un approccio alternativo consiste nell'utilizzare i **container** o **contenitori**, che si possono considerare come delle **virtual machine** leggere che contengono solo lo stretto indispensabile (**dipendenze**, file di configurazione e così via) per eseguire il servizio in questione.
Il miglior sistema di implementazione di questo metodo è **Docker**, usato anche in questo articolo. Anche se **Docker-cockpit** non è più supportato in **Cockpit**, funziona ancora molto bene. Per scaricarlo e installarlo usate i comandi:

```
$ wget https://launchpad.net/ubuntu/+source/cockpit/215-1~ubuntu19.10.1/+build/18889196/+files/cockpit-docker_215-1~ubuntu19.10.1_all.deb
$ sudo apt install ./cockpit-docker_215-1~ubuntu19.10.1_all.deb
```

Dopo averlo installato, **sloggatevi** e rientrate in **Cockpit**, dove vedrete **Docker Containers** nella parte sinistra dello schermo. Selezionatelo per scoprire le gioie di utilizzare e gestire i contenitori senza dover armeggiare da riga di comando.
Quando fate click sul vostro account utente vedete anche comparire una nuova opzione sotto **Accounts**: **Container Administrator**. Aggiunge gli utenti selezionati al **gruppo Docker** per consentirgli di eseguire l'applicazione da riga di comando senza bisogno di usare **sudo**. Ora che avete impostato il cuore del vostro server NAS, fate un altro backup con **Timeshift** prima di procedere a godervi il meritato frutto del vostro lavoro! LXP

TIP

Cercate nel BIOS le impostazioni per il risparmio energetico per spegnere i bus SATA e PCIe quando non sono usati, una scelta comune per i server.

Name	Description	State
fwupd-refresh	Refresh fwupd metadata and update motd	failed
accounts-daemon	Accounts Service	active (running)
apparmor	Load AppArmor profiles	active (exited)
apport-autoreport	Process error reports when automatic reporting is enabled	inactive (dead)
apport-forward@	apport-forward@.service Template	
apport	LSB: automatic crash report generation	active (exited)
apt-daily-upgrade	Daily apt upgrade and clean activities	inactive (dead)
apt-daily	Daily apt download activities	inactive (dead)
atd	Deferred execution scheduler	active (running)

Anche se non è più ufficialmente supportato, **Docker-cockpit** funziona molto bene

ASYNCIO

Creare codice asincrono in Python

Preparatevi a usare il multitasking come non lo avete mai fatto prima con le mille applicazioni di questo modulo

Considerate di avere una pagina Web che deve caricare dati da diversi server e che alcuni di essi non rispondano in tempi brevi. Analogamente immaginate ora di star scrivendo uno **script** in **Python** che deve interagire con più macchine o server. Volete che si fermi ad aspettare un dato server o che continui le sue operazioni monitorandolo? In generale, la seconda strategia è la migliore per ottimizzare le risorse. Su una pagina Web, potete risolvere il problema scrivendo **codice JavaScript asincrono**, mentre in Python la soluzione è la **programmazione asincrona**.
In questo articolo per adottarla vengono utilizzati **Python 3** e il **modulo asyncio**, una libreria per scrivere **codice concorrente** che segue il paradigma **async/await**. Asyncio fa parte delle librerie standard di Python dalla versione **3.4** ma qui si assume che usiate la **3.8** o un'edizione successiva.

Le basi del modulo asyncio

Lo scopo della libreria asyncio è che un solo **thread** esegua un **ciclo di eventi** (**event loop**). In questo ciclo, che è il **core** di ogni applicazione asyncio, viene cambiato **task** quando il programmatore scrive la **keyword** `await`, quindi la decisione non viene affidata allo **scheduler** del sistema operativo sulla base dei segnali **await**. Asyncio non risulta quindi efficiente per i task con carichi intensivi sulla **CPU**, perché non gestisce più **thread** in parallelo, ed è più adatto per operazioni di **input/output** asincrone come lavorare con **socket di rete**, per via di come sono implementati in Python (sono scritti in **C** e usano risorse diverse da Python).
In questa sede sono simulati dei ritardi con chiamate che mettono una funzione in **sleep**. Asyncio fa parte della libreria Python 3, quindi non serve installare altre librerie per usarlo e basta includere uno **statement** `import asyncio` nel codice. Prima di procedere, chiariamo dei presupposti di base. Le operazioni asincrone restituiscono subito la promessa che daranno il loro

TIP

Trovate la documentazione ufficiale di asyncio all'indirizzo https://docs.python.org/3/library/asyncio.html. Ci sono ulteriori informazioni anche su https://docs.python.org/3/library/asyncio-dev.html e https://realpython.com/async-io-python.

```
Task 5 done
Task 6 done
Task 7 done
Task 8 done
Task 9 done
Non sequentially:
Task 1 done
Task 2 done
Task 3 done
Task 6 done
Task 4 done
Task 5 done
Task 9 done
Task 7 done
Task 8 done
→ code
```

```
Task 5 done
Task 6 done
Task 7 done
Task 8 done
Task 9 done
Non sequentially:
Task 4 done
Task 5 done
Task 6 done
Task 1 done
Task 3 done
Task 7 done
Task 8 done
Task 2 done
Task 9 done
→ code
```

L'**output** di **order.py** che illustra l'**esecuzione sequenziale** e **asincrona** di funzioni e **coroutine**, rispettivamente. Le coroutine sono eseguite in ordine casuale

risultato quando possibile. Un oggetto si definisce **awaitable** se si può usare in un'espressione `await`.
I tre principali tipi di oggetti awaitable sono **coroutine**, **task** e **future**. Le prime sono una forma più generica di **subroutine**. Mentre queste ultime sono invocate in un punto ed escono in un altro, le coroutine si possono invocare, far uscire e far continuare in più punti e sono implementate con statement `async def`. Potete considerare il codice che segue, salvato come **asyncIOhw.py**, come la versione asyncio di "Ciao mondo!":

```
import asyncio
async def main():
    print('Ciao ...')
    await asyncio.sleep(1)
    print('... mondo!')
asyncio.run(main())
```

Ricordate che asyncio ha la sua versione speciale di `sleep()`, che viene richiamata come `asyncio.sleep()`, che è una **funzione non bloccante**. In pratica, `asyncio.sleep()` sospende il task corrente che ha richiamato la funzione, ma permette a tutti gli altri di operare e non blocca l'intero script. Il codice printa "Ciao ...", attende un secondo per la chiamata

`asyncio.sleep(1)` e poi procede con "... mondo!". Come succede? Create la funzione `main()` che eseguite richiamando `asyncio.run()`.

Ordine di esecuzione

Risulta quindi chiaro che non dovete fare assunzioni su quale sarà l'ordine di esecuzione delle coroutine quando usate asyncio. Questo principio si applica anche alla programmazione concorrente e parallela. Questa caratteristica è presente in **order.py**, il cui codice più importante è:

```
async def task_coroutine(pid):
    await asyncio.sleep(random.randint(0, 2))
    print('Task %s eseguito' % pid)
async def concurrently():
    tasks = [task_coroutine(i) for i in range(1, 10)]
    await asyncio.gather(*tasks)
asyncio.run(concurrently())
```

Qui succedono tre cose. La prima è che viene creata la coroutine `task_coroutine()` che sta in sleep per una quantità casuale di tempo. La seconda è che viene definita un'altra coroutine di nome `concurrently()` che genera dieci coroutine `task_coroutine()`. I loro risultati sono raccolti con `asyncio.gather()`. Viene infine eseguito `concurrently()` usando `asyncio.run()`. Tutte le funzioni definite come asincrone dovrebbero essere eseguite usando le funzionalità della libreria asyncio.
La schermata nella prima pagina di questo articolo mostra l'output di **order.py** quando viene eseguito due volte. L'esecuzione sequenziale genera sempre lo stesso risultato perché le funzioni sono appunto eseguite nello stesso ordine (sequenziale), mentre l'esecuzione asincrona genera ogni volta un output diverso perché le coroutine sono eseguite in ordine casuale.

Processare valori multipli

Vediamo un esempio in cui vengono create più coroutine e si raccolgono i valori restituiti. Il nome dello script è **results.py**. La schermata qui in alto mostra il suo codice in Python. L'elemento più importante è il suo uso per l'elaborazione di un ciclo di eventi (event loop), creato con una chiamata a `asyncio.get_event_loop()`. C'è poi l'utilizzo di `run_until_complete()` per eseguire il numero di **runner** necessari a svolgere i compiti desiderati. Il parametro di `runner()` è una lista di numeri interi. Si passa quindi alla coroutine `runner()`, che è quella che esegue il numero richiesto di coroutine `fibo()` e raccoglie i risultati. La lista di coroutine è creata con l'aiuto di `asyncio.create_task()`. Il parametro di `asyncio.wait()` è la lista delle coroutine che stanno per essere eseguite. I valori restituiti da `asyncio.wait()` sono due **future** (vedi seguito). Al termine dell'elaborazione, i risultati sono raccolti con lo statement `results = [res.result() for res in done]`. L'output di **results.py** è:

```
$ ./results.py
Totale risultati: 7
Risultati in sospeso: 0
[1, 1, 2, 3, 8, 55, 610]
```

L'ultima riga mostra l'output ordinato (dato che l'ordine di esecuzione non si può determinare), mentre la prima riga **printa** il numero totale di risultati raccolti e la seconda che ce ne sono **0** in sospeso.

```
6  async def fibo(value):
7      if value < 0:
8          raise "Negative number"
9
10     a, b = 0, 1
11     for i in range(2, value+1):
12         a, b = b, a+b
13
14     return b
15
16 async def runner(values: List[int]):
17     requests = [asyncio.create_task(fibo(number)) for number in values]
18     done, pending = await asyncio.wait(requests)
19
20     results = [res.result() for res in done]
21     print('Total results: {}'.format(len(results)))
22     print('Pending results: {}'.format(len(pending)))
23     print(sorted(results))
24
25 if __name__ == '__main__':
26     loop = asyncio.get_event_loop()
27     loop.run_until_complete(runner([1, 10, 2, 3, 4, 15, 6]))
28     loop.close()
```

Il codice di **results.py** in cui sono eseguite diverse **coroutine** e ne vengono raccolti i risultati. Qui ogni coroutine calcola un numero della **successione di Fibonacci**, ma potete fargli fare ciò che volete

Come usare task e future

Un task è un modo per **schedulare** l'esecuzione di una coroutine, mentre un future è un riferimento indiretto a un risultato che arriverà. Potete chiedere a un future di inviare la sua risposta quando è pronta. I task sono illustrati in **tasks.py**, che potete vedere nella schermata a **pag. 72**, mentre i future sono mostrati in **futures.py**, il cui codice più importante è **futures.py**:

```
async def main():
    future = asyncio.Future()
    await asyncio.ensure_future(fibo(future, 10))
    print(future.result())
```

Qui, prima di tutto viene definito un oggetto future. La chiamata `asyncio.ensure_future()` converte poi la coroutine `fibo()` in un oggetto future.
Viene quindi printato il risultato del future con `print(future.result())`. Tenete a mente che `ensure_`

» CONCORRENTI E PARALLELI

Molti confondono i concetti di **concorrenza** e di **parallelismo**, ritenendoli sinonimi. Con quest'ultimo vengono eseguite simultaneamente diverse entità di un certo tipo, mentre la concorrenza è un modo per strutturare i componenti in modo che possano essere eseguiti indipendentemente. Solo dopo aver creato i vostri componenti software in modo concorrente potete eseguirli in sicurezza in parallelo, quando e se il vostro sistema operativo e il vostro hardware lo consentono. Il linguaggio di programmazione **Erlang** (**www.erlang.org**) ha iniziato a farlo molto tempo fa, parecchio prima che le **CPU** avessero **core** multipli e i computer godessero di moltissima **RAM**. Con un **design** valido, aggiungere entità concorrenti rende più veloce l'intero sistema, consentendo l'esecuzione in parallelo.
Lo sviluppatore ha il compito di considerare la concorrenza nella fase di progettazione di un sistema e beneficerà della potenziale esecuzione in parallelo dei suoi componenti. Chi programma non deve quindi concentrarsi sul concetto di parallelismo, ma suddividere il codice in componenti indipendenti che risolvono il problema iniziale insieme. Anche se non potete eseguire le funzioni in parallelo su una macchina **UNIX**, un design concorrente valido migliora comunque la struttura e la mantenibilità dei vostri programmi.

TIP

Il processo di sviluppo di un programma dovrebbe tenere in considerazione l'uso di asyncio nella progettazione. Il risultato sono programmi più efficienti e veloci, ma bisogna ricordare che non tutto si può eseguire in modo concorrente.

`future()` inizia a eseguire la coroutine, che viene passata come parametro, non appena l'event loop viene ripreso. Ciò che qui non si vede è che `fibo()` restituisce il valore desiderato usando una chiamata `future.set_result()`, che è il metodo migliore con i future. Fate riferimento al codice completo di **futures.py** per ulteriori dettagli. Nella schermata in basso vedete l'output di **tasks.py** e **futures.py** oltre al codice di **tasks.py**. Lo script **futures.py** restituisce un singolo risultato mentre **tasks.py** esegue più task usando `asyncio.create_task()`. Ancora una volta, potete ricevere il risultato di un task con il metodo `result()`.

Calcolare la successione di Fibonacci

Calcolate ora la **successione di Fibonacci** in modo concorrente usando `asyncio.gather()`, responsabile di raccogliere tutti i risultati. `asyncio.gather()` attende finché tutti gli oggetti disponibili sono finiti e poi raccoglie i risultati. Il codice più importante di **fibo.py** è l'implementazione di `main()` che contiene la chiamata `asyncio.gather()`:

```
async def main():
    result = await asyncio.gather(
        fibo("A", -1),
        fibo("B", 3),
        fibo("C", 4),
        fibo("D", 5),
        fibo("E", 6),
        return_exceptions=True
    )
    print(result)
```

Qui viene fatto lo **scheduling** di cinque chiamate a funzioni da eseguire concorrentemente. La funzione `asyncio.gather()` si aspetta che le **awaitable** siano passate come **argomenti posizionali**. L'uso di `return_exceptions=True` in `asyncio.gather()` permette inoltre di raccogliere le eccezioni che possono verificarsi nel processo. In questo caso l'eccezione interviene con la chiamata `fibo("A", -1)` perché viene passato un valore negativo a `fibo() function`. Malgrado l'eccezione, le altre funzioni continuano a operare. Potete vedere l'output of **fibo.py** nella schermata in alto a **pag. 73**. Le righe del risultato contengono

```
→ code cat tasks.py
#!/usr/bin/env python3

import asyncio
import random

async def this_is_a_task():
    print("Task execution")
    temp = random.randint(0,100)
    return temp

async def main():
    for i in range(0, 5):
        task = asyncio.create_task(this_is_a_task())
        await task
        print(task.result())

asyncio.run(main())
→ code ./tasks.py
Task execution
95
```

In questa schermata potete vedere il codice di **tasks.py** e l'**output** di quest'ultimo e di **futures.py** che illustrano rispettivamente l'uso di **task** e **future** in **asyncio**

solo il testo di `fibonacci(x)` mentre quelle che contengono `Compute` mostrano l'avanzamento dei calcoli. Tutte devono calcolare `fibonacci(2)` per trovare il numero desiderato della successione di Fibonacci, ma solo l'ultima deve calcolare `fibonacci(6)`. L'**output** mostra infine che il tempo della CPU è suddiviso tra le coroutine disponibili.

Quando non si schedula una coroutine

Se vi dimenticate di fare lo **scheduling** di una coroutine, Python se ne accorge e visualizza degli avvisi, come viene illustrato in **never.py**. In questo estratto di codice:

```
async def main():
    test()
```

viene richiamato `test()`, che è implementato altrove, senza che sia stato **schedulato**. L'output di **never.py** sarà:

```
$ ./never.py
./never.py:10: RuntimeWarning: coroutine 'test' was
never awaited
  test()
RuntimeWarning: Enable tracemalloc to get the object
allocation traceback
main() is about to return.
```

Viene mostrata l'origine del problema, ossia che il codice non aspetta mai che la coroutine `test()` finisca quindi non è mai schedulata per l'esecuzione. Ci sono due soluzioni che potete adottare:

```
asyncio.create_task(test())
await test()
```

Con entrambe le tecniche potete fare lo scheduling dell'esecuzione di `test()`.

Sviluppare un server Echo

Passate ora a sviluppare un **server Echo** in Python che usa asyncio. La schermata in basso nella prossima pagina ne mostra il codice. La parte più importante di **echo.py** è:

```
server = await asyncio.start_server(handle, '', PORT)
addr = server.sockets[0].getsockname()
```

La parte più significativa è la funzione `asyncio.start_server()` che avvia un **socket server** usando i parametri specificati. Poiché il secondo è `''`, il server sarà in ascolto di tutte le interfacce di rete disponibili, il che può essere un rischio per la sicurezza se è esposto a Internet. Sostituite `''` con `'127.0.0.1'` per accettare connessioni solo dalla macchina locale. Il primo parametro è la funzione che viene eseguita ogni volta che un nuovo **client** è collegato al server, per gestire la connessione. Quella **funzione handler** ha due parametri, che sono il **reader** e il **writer**, rispettivamente, e sono usati per interagire con i client. Le coroutine **handle(reader, writer)** sono create per interagire con i client fino a che un client invia la **keyword STOP**, che è definita dal codice e chiude la connessione. Le altre connessioni continuano però a funzionare. Anche se un client ha un'interazione lenta con il server, inoltre, gli altri non subiscono ritardi. Quando è in esecuzione e interagisce con il client, **echo.py** genera il seguente output:

```
$ ./echo.py
Serving on ('::', 1234, 0, 0)
```

```
Task B: Compute fibonacci(3)...
Task C: Compute fibonacci(3)...
Task D: Compute fibonacci(3)...
Task E: Compute fibonacci(3)...
Task B: fibonacci(3) = 2
Task C: Compute fibonacci(4)...
Task D: Compute fibonacci(4)...
Task E: Compute fibonacci(4)...
Task C: fibonacci(4) = 3
Task D: Compute fibonacci(5)...
Task E: Compute fibonacci(5)...
Task D: fibonacci(5) = 5
Task E: Compute fibonacci(6)...
Task E: fibonacci(6) = 8
[TypeError('exceptions must derive from BaseException')
→ code
```

La funzione **asyncio.gather()** permette di raccogliere i risultati di tutte le **coroutine**, incluse le possibili eccezioni

```
Machine ('192.168.1.7', 42750) just connected.
Machine ('127.0.0.1', 51288) just connected.
('127.0.0.1', 51288) closed the connection.
Machine ('127.0.0.1', 51291) just connected.
Machine ('127.0.0.1', 51292) just connected.
('127.0.0.1', 51292) closed the connection.
('127.0.0.1', 51291) closed the connection.
('192.168.1.7', 42750) closed the connection.
```

La brevità, la semplicità e la leggibilità di **echo.py** rendono asyncio ideale per creare **utility TCP/IP**, soprattutto se considerate che questi server hanno frequenti ritardi dovuti a problemi di rete o interazioni lente con i client. Trovate ulteriori dettagli (in inglese) sulla creazione di server di **streaming** con l'utilizzo di asyncio all'indirizzo **https://bit.ly/38Ga6Qw.**

È davvero più veloce?

Per verificare che asyncio effettivamente velocizzi il lavoro quando ci sono dei ritardi, provate a calcolare i numeri della successione di Fibonacci con e senza il suo utilizzo. Il codice di **multiFutures.py** presenta la versione asyncio mentre **serialFibo.py** mostra quella sequenziale. La logica del primo è:

```
async def main():
    tasks = []
    for i in range(15):
        tasks.append(asyncio.ensure_future(random_
delay(i)))
    await asyncio.gather(*tasks)
```

La funzione **main()** definisce il numero desiderato di coroutine usando **asyncio.ensure_future()** e le mette in una lista di nome **tasks**, dopo di che **asyncio.gather(*tasks)** esegue la lista di coroutine e attende che terminino. Fate riferimento al codice di **multiFutures.py** per ulteriori dettagli. I risultati di questi due script in Python rivelano che la versione asincrona è sempre più veloce di quella sequenziale, soprattutto perché i tempi di **sleep** (ritardo) di quest'ultima vengono sommati mentre quelli della versione asincrona sono condivisi. Il codice di **serialFibo.py** è però più facile da leggere, in particolare se non avete familiarità con asyncio. La versione asincrona usa infine i future e **asyncio.gather()**, il che è comune quando si usa asyncio. Anche se non è presentato in questo articolo, la stessa tecnica si può usare quando si interroga un **database server** e si è in attesa di una risposta. Dato che la risposta a una **query** potrebbe richiedere un certo tempo, in base alla natura dell'**interrogazione** e al carico del database server, usare asyncio risulta una buona scelta. Il modo migliore per imparare asyncio e le sue particolarità consiste nell'utilizzarlo per creare programmi. Cominciate da progetti piccoli e facili, sfruttando anche quanto appreso con gli esempi di questo articolo, prima di continuare con lavori più complessi. Una volta che vi siete abituati a come funziona, applicate questi principi a risolvere i problemi del mondo reale. Lo sforzo necessario per utilizzare le funzioni di asyncio e scrivere un po' di codice in più per usare le operazioni asincrone sarà ripagato dalla maggior velocità dei programmi, che in alcuni contesti può risultare significativa. LXP

» GLOBAL INTERPRETER LOCK (GIL)

Dato che **Global Interpreter Lock** (**GIL**) fa parte di **Python** e del suo funzionamento, è utile conoscerlo meglio. Si tratta di un **mutex** (**mutual exclusion variable**) che permette solo a un singolo **thread** di avere il controllo dell'**interprete** di Python, ossia fa sì che in ogni dato momento possa essere eseguito un solo thread. In un mondo di **CPU** e **core** multipli, può risultare un ostacolo. Ci sono però dei motivi per la sua presenza. In primo luogo serve a evitare gli **stalli** o **deadlock**, che sono un problema serio nella programmazione **multi-thread**. Alla nascita di Python, inoltre, i computer non avevano più core quindi GIL non era in questione e in seguito non è stato rimosso perché farlo creerebbe una serie di problemi di **retrocompatibilità**, il che in un linguaggio popolare come Python avrebbe conseguenze molto gravi. Se volete utilizzare i thread in Python, ci sono tre moduli che potete usare: **low level _thread** (**https://docs.python.org/3/library/_thread.html**), **threading** (**https://docs.python.org/3/library/threading.html**) e **multiprocessing** (**https://docs.python.org/3/library/multiprocessing.html**). Approfondire ulteriormente questo tema esula dagli scopi di questo articolo ma, a causa del funzionamento di GIL e delle restrizioni che comporta per il codice, usare più thread spesso non velocizza i vostri programmi. Un motivo in più per provare la **programmazione asincrona** sfruttando **asyncio**.

```
11     while True:
12         data = await reader.read(100)
13         message = data.decode().strip()
14         writer.write(data)
15         await writer.drain()
16         if message == "STOP":
17             writer.write(b'Bye!\n')
18             message = f"{addr!r} closed the connection."
19             print(message)
20             break
21 
22     writer.close()
23 
24 async def main():
25     # Listen on all interfaces and port number PORT
26     server = await asyncio.start_server(handle, '', PORT)
27     addr = server.sockets[0].getsockname()
28 
29     print(f'Serving on {addr}')
30     async with server:
31         await server.serve_forever()
```

L'implementazione di un **server Echo** con **asyncio** in **echo.py**. Gran parte del lavoro è svolta da **asyncio.start_server()** oltre che dalla funzione che gestisce le connessioni al **socket**, qui chiamata **handle()**

PYGAME

Crediti: https://raytomely.itch.io/run-and-gun

Zombie Runner risorge in Python

Potete ricreare facilmente un classico dei videogame utilizzando un motore di gioco con interfaccia utente grafica

Da **Resident Evil** a **Left 4 Dead** e **The Last Of Us**, i giochi di zombie hanno sempre il loro perché. Il classico **Zombie runner** è uno **sparatutto** in cui il personaggio del giocatore si muove in un mondo bidimensionale evitando il contatto con i non morti e sparandogli. In questo tipo di titolo negli anni Settanta il protagonista era fisso e il mondo di gioco gli si muoveva attorno, mentre nella decade successiva il giocatore si muoveva su un singolo asse direzionale. Negli anni Novanta, invece, il movimento del personaggio principale era limitato da specifici percorsi sullo schermo. In queste pagine creerete uno sparatutto in cui il giocatore può esplorare liberamente gli ambienti, uccidendo zombie e nemici nel frattempo. Il mondo è bidimensionale, ma la giocabilità è moderna.

Installare Python e pyGame

Per questo progetto dovete avere **Python3** installato. Eseguite `sudo apt-get install python3` se avete una distribuzione basata su **Debian**. Potete quindi usare **Python IDLE** per scrivere il codice. Ricordatevi di selezionare **File > New File** all'apertura e di scrivere sempre nella finestra **Script** e non in quella **Shell** che si apre automaticamente. Per salvare o testare il gioco basta premere **F5**, la prima volta specificando dove archiviare il file. Non siete però obbligati a usare IDLE: potete utilizzare un qualsiasi editor per creare un file Python, purché lo salviate con l'estensione **.py**. Per eseguirlo dovete solo scrivere `python3 nomefile.py` in un **terminale**. **PyGame** è un ottimo **toolset** che integra un'ampia serie di metodi di programmazione per i videogame incorporabili nel vostro codice. Utilizzarlo vi permette di concentrarvi sulla progettazione del gioco anziché sul reinventare la ruota. Installatelo con `pip3 install pygame`.

Il motore Run n Gun

Il codice e la grafica sono basati interamente sul **motore di gioco Run and Gun** di Ray **Tomely** (**https://raytomely.itch.io/run-and-gun**). Scaricate ed estraete il file **.rar**. Passate quindi a esaminare il codice di **run_n_gun.py** per capirne il funzionamento. Il codice

```
import pygame,sys,pickle
from pygame.locals import *
from runngun_classes import *
```

importa le classi del **modulo di Run and Gun** che avete scaricato. Qui viene utilizzata la **programmazione orientata agli oggetti**, in modo da poter ottimizzare il codice importando e riciclando quello ripetuto. Sono anche importati i moduli di PyGame che di fatto operano come un motore di gioco, in modo che possiate concentrarvi sulla posizione e i valori dei vostri **sprite** invece di dover insegnare a Python come si disegna un oggetto sullo schermo. Quando si programma con un **linguaggio di alto livello** come Python, è sempre utile sfruttare tutti gli ausili disponibili senza riscrivere codice già esistente. Inizializzate poi PyGame e dichiarate le impostazioni dello schermo. **640 x 480** è una definizione standard per la bassa risoluzione, ma potrete ridimensionare la finestra e impostarla a schermo intero.

```
pygame.init()
screen = pygame.display.set_mode((640, 480),)
SCREEN_WIDTH=640
```

Ecco l'eroe del gioco, a caccia di guai sin dall'inizio

TIP

Potete modificare il testo del gioco in dialog.py, cambiando non solo il tipo di carattere, ma anche le sue dimensioni, il colore di sfondo, quello dei bordi e la posizione del testo.

```
SCREEN_HEIGHT=480
pygame.display.set_caption("run n gun")
```
Impostate quindi i colori e il **clock** del gioco, che controlla il numero di **fotogrammi al secondo**.
```
BLACK=pygame.color.THECOLORS["black"]
WHITE=pygame.color.THECOLORS["white"]
CLOCK=pygame.time.Clock()
```
Puntate quindi al percorso della cartella dei **tileset** degli sprite scaricati in precedenza.
```
PATTH="run_n_gun_data/"
sprite_sheet=pygame.image.load(PATTH+"rambito.
png").convert()
shadow=pygame.image.load(PATTH+"rambito_shadow.
png").convert()
tileset=pygame.image.load(PATTH+"tileset.png").
convert()
alpha_color=sprite_sheet.get_at((0,0))
sprite_sheet.set_colorkey(alpha_color)
shadow.set_colorkey(alpha_color)
shadow.set_alpha(100)
```
Ci sono lo sprite del personaggio del giocatore, le ombre e il tileset del mondo. Impostate anche il **colore alpha**, che consente di usare le **trasparenze**. Dovete quindi definire una posizione per questi sprite, indicando dove compaiono nel gioco il personaggio principale, i nemici e gli altri elementi:
```
player_pos=[320,240]
player_box=[player_pos[0]+48,player_pos[1]+70]#[45,55]
player_box_width=30#40
player_box_height=30#40
player_direction="up"
move_speed=10
scrol_box=[210,130]
scrol_box_width=220
scrol_box_height=220
scrol_x=0
scrol_y=0
scroling=0
up_limit=576
gun=Gun()
enemy_group=Enemy.group
active_enemy=[]
active_solid_tile=[]
```
Ora che tutte le variabili sono state impostate, potete iniziare a scrivere il codice della logica.
```
with open('level.lvl', 'rb') as file:
    file_loader=pickle.Unpickler(file)
    level_tile=file_loader.load()
    level_tile_pos=file_loader.load()
    level_solid_tile_index=file_loader.load()
    file_loader.load()
    for pos in file_loader.load():
        Enemy(pos)
top_left_limit=min(level_tile_pos)
bottom_right_limit=max(level_tile_pos)
for i in range(len(level_tile_pos)):
    if top_left_limit==level_tile_pos[i]:
        top_left_limit=i
    elif bottom_right_limit==level_tile_pos[i]:
        bottom_right_limit=i
```
Importate il livello del gioco da un file esterno, il che vi dà molta più libertà che integrare i livelli nel codice. In questo modo, infatti, potete cambiare il loro contenuto in qualsiasi momento senza andare a modificare il programma. Potete inoltre consentire agli utenti di alterare e importare i livelli senza bisogno di saper programmare. All'interno di **level.lvl**, **level_2.lvl** o **level_3.lvl** potete vedere **array** con **valori RGB**. Il motore di gioco Run and Gun include un **editor dei livelli**, quindi non avete bisogno di crearli o modificarli manualmente, ma potete farlo se volete. Per vedere il livello dovete disegnare le caselle e gli sprite:
```
sheet_width=sprite_sheet.get_width()
sheet_height=sprite_sheet.get_height()
hor_cells=5
ver_cells=8
scale=4
sprite_sheet = pygame.transform.scale(sprite_sheet,
(sheet_width*scale, sheet_height*scale))
```

Gli zombie diventano rossi quando gli sparate

TIP

Qui sono usate immagini Free and Open Source per il gioco, ma potete scaricare altri pacchetti di sprite o crearne di vostri. Nella cartella run_n_gun\run_n_gun_data troverete i tileset per le mappe dei livelli e gli sprite animati.

» RILEVAMENTO DELLE COLLISIONI

Il sistema di rilevamento trattato è semplice, ma funzionale. All'interno di **runngun_classes.py** c'è una classe creata specificamente per calcolare se due forme collidono:
```
def collide(point,rect):
    collided=0
    if point[0]>=rect[0] and point[0]<rect[0]+rect[2] \
    and point[1]>=rect[1] and point[1]<rect[1]+rect[3]:
        collided=1
    return collided
```
Dati un punto e un rettangolo, se i punti agli angoli collidono, la **variabile booleana** è impostata su **true**. Il discorso si complica se si fa la rilevazione delle collisioni per dei cerchi:
```
def circle_collision(circle1,circle2):
    collision=0
    c1=circle1.center
    c2=circle2.center
    length=math.sqrt((c1[0]-c2[0])**2+(c1[1]-c2[1])**2)
    if circle1.radius+circle2.radius>length:
        collision=1
    return collision
```
Qui viene misurato il raggio dal centro e si calcola la distanza con una funzione per la radice quadrata. Se vi viene in mente un metodo per gestire il problema in modo più semplice e immediato, scriveteci in redazione!

```
shadow = pygame.transform.scale(shadow, (sheet_
width*scale, sheet_height*scale))
cell_width=int(sprite_sheet.get_width()/hor_cells)
cell_height=int(sprite_sheet.get_height()/ver_cells)

tile_width=16*scale
tile_height=16*scale
tileset_width=tileset.get_width()
tileset_height=tileset.get_height()
tileset=pygame.transform.scale(tileset,(tileset_
width*scale,tileset_height*scale))

up_frame_pos_y=0
down_frame_pos_y=cell_height*4
right_frame_pos_y=cell_height*2
left_frame_pos_y=cell_height*6
anim_frame=1
max_anim_time=3
anime_time=0
frame_pos_x=0
frame_pos_y=up_frame_pos_y

up = down = left = right = False
```

Si passa quindi a un semplice **rilevamento delle collisioni**. Un **binario** determina se due coordinate si incrociano in qualche punto:

```
def colide(x_pos,y_pos,width,height,x_pos2,y_pos2):
   colision=0
   if x_pos2>x_pos and x_pos2<x_pos+width and y_
pos2>y_pos and y_pos2<y_pos+height:
      colision=1
   return colision
```

Bisogna poi impostare un **ciclo**, in modo che l'azione non si interrompa. Per farlo, usate la **velocità di clock**, che viene fissata a 30 fotogrammi al secondo e assicura un'esperienza di gioco stabile.

```
pygame.key.set_repeat(400, 30)

while True:
   CLOCK.tick(30)
   scrol_x=0
   scrol_y=0
   scroling=0
   active_enemy=[]
   active_solid_tile=[]
```

Un giretto nel verde non è completo senza un morto vivente a cui sparare!

Siamo inseguiti da mostri da ogni direzione!

Qualsiasi altro elemento aggiungiate apparterrà a questo **loop**. Il **ciclo while** mantiene il gioco in esecuzione. Metteteci altri elementi.

```
   for event in pygame.event.get():
      if event.type == QUIT:
         pygame.quit()
         sys.exit()

      if event.type == KEYUP:
         anim_frame=1
         frame_pos_x=0
         anime_time=0
         up = down = left = right = False
```

Qui ci si mette in attesa di un evento, in questo caso la pressione di un tasto o il comando di chiusura. Se viene inserito quest'ultimo, uscite da Pygame e chiudete il programma. Se, invece, viene premuto un tasto, si anima il personaggio. Più nel dettaglio:

```
   keys = pygame.key.get_pressed()
   if keys[K_UP]:
      up=True
      gun.direction="up"
      player_pos[1]-=move_speed
      if player_pos[1]<0:
         player_pos[1]=0
      frame_pos_y=up_frame_pos_y
      anime_time+=1
      if anime_time>=max_anim_time:
         anime_time=0
         if anim_frame<hor_cells:
            anim_frame+=1
            frame_pos_x+=cell_width
         else:
            anim_frame=2
            frame_pos_x=cell_width
      if up:
         if level_tile_pos[top_left_limit][1]<=-move_speed:
            if player_box[1]<scrol_box[1]:
               player_pos[1]+=move_speed
               scrol_y+=move_speed
               scroling=1
```

Con questo codice è stata creata una variabile di nome **keys** che monitora la pressione dei tasti. Poi è stato programmato uno **statement IF** per cercare la freccia in alto (**K_UP**), che sposta il personaggio verso la parte superiore dello schermo, modificando l'**asse Y**. Gli incrementi positivi spostano verso l'alto e quelli

©italia_news

Ricordatevi di guardarvi sempre alle spalle!

negativi verso il basso. Prendete il codice e modificatelo per il tasto freccia in basso (**K_DOWN**):

```
elif keys[K_DOWN]:
  down=True
  gun.direction="down"
  player_pos[1]+=move_speed
  if player_pos[1]>SCREEN_HEIGHT-cell_height:
    player_pos[1]=SCREEN_HEIGHT-cell_height
  frame_pos_y=down_frame_pos_y
  anime_time+=1
  if anime_time>=max_anim_time:
    anime_time=0
    if anim_frame<hor_cells:
      anim_frame+=1
      frame_pos_x+=cell_width
    else:
      anim_frame=2
      frame_pos_x=cell_width
  if down:
    if level_tile_pos[bottom_right_limit]
[1]>=(SCREEN_HEIGHT+move_speed)-tile_height:
      if player_box[1]+player_box_height>scrol_
box[1]+scrol_box_height:
        player_pos[1]-=move_speed
        scrol_y-=move_speed
        scroling=1
```

Per spostare il personaggio a sinistra o a destra si usa la stessa procedura, ma sull'**asse X**; gli incrementi positivi lo portano a destra e quelli negativi a sinistra.

Pronti al combattimento!

Con i **tasti direzionali** definiti, aggiungete uno **statement IF** per monitorare il **tasto F**, usato per sparare:

```
if keys[K_f]:
  gun.fire=1
```

Semplicemente si utilizza un **valore booleano** per indicare che è stato sparato un colpo. Bisogna sapere verso quale direzione è puntata l'arma:

```
if up or down or left or right:
  player_box=[player_pos[0]+48,player_pos[1]+70]
```

E anche indicare da dove spara, ossia la posizione del personaggio del giocatore:

```
if gun.fire:
  gun.update(player_pos)
```

Ora che potete controllare il personaggio, spostarvi per il mondo di gioco e sparare, servono dei nemici. Per prima cosa create una scorta infinita di zombi:

» L'EDITOR DEI LIVELLI

Il motore di gioco utilizzato in questo articolo offre un editor dei livelli completo, quindi potete essere creativi anche se non vi piace programmare. Eseguite **world_editor.py** per lavorare con la mappa del livello predefinito. È sempre meglio modificare livelli esistenti prima di crearne di nuovi. Ecco alcuni controlli utili:
O per cambiare **tileset**;
S per alternare la modifica e la selezione delle schermate;
L per andare all'area successiva e **P** per tornare alla precedente;
N per marcare la posizione di inizio;
Quando siete in modalità di modifica (**edit**), **O** cambia **actor**, **T** cambia il tipo di casella e **C** ripulisce il livello. La categoria **actor** comprende nemici, personaggi non giocanti e **sprite** degli oggetti (per esempio le nuvole). Le caselle standard sono evidenziate da un rettangolo arancione mentre quelle di passaggio da un'area all'altra (dette **gate**) hanno un rettangolo blu. Potete spostare i **tileset** con i **tasti freccia** e anche fare click sulle caselle e trascinarle dove vi servono.

```
for enemy in enemy_group:
  if not enemy.death:
    if scroling:
      enemy.pos[0]+=scrol_x
      enemy.pos[1]+=scrol_y
    if enemy.pos[0]>=-128 and enemy.
pos[0]<SCREEN_WIDTH+128 \
    and enemy.pos[1]>=-128 and enemy.
pos[1]<SCREEN_HEIGHT+128:
      active_enemy.append(enemy)
    else:
      enemy.blit_ok=0
```

Vanno quindi disegnati sia le caselle nei punti corretti del livello sia il personaggio del giocatore mentre si sposta attraverso di esso nelle quattro direzioni. Il brano di codice è molto lungo ma vale la pena di esaminarlo:

```
screen.fill(BLACK)
for i in range(len(level_tile_pos)):
  if scroling:
    level_tile_pos[i][0]+=scrol_x
    level_tile_pos[i][1]+=scrol_y
  if level_tile_pos[i][0]>=-tile_width and level_tile_pos[i]
[0]<SCREEN_WIDTH \
  and level_tile_pos[i][1]>=-tile_height and level_tile_
pos[i][1]<SCREEN_HEIGHT:
```

È meglio cercare di radunare i nemici che attaccarli uno alla volta

```
            screen.blit(tileset,level_tile_pos[i],(level_tile[i]
[0],level_tile[i][1],tile_width,tile_height))
    for i in level_solid_tile_index:
        if level_tile_pos[i][0]>=-tile_width and level_tile_
pos[i][0]<SCREEN_WIDTH+tile_width \
        and level_tile_pos[i][1]>=-tile_height and level_tile_
pos[i][1]<SCREEN_HEIGHT+tile_height:
            active_solid_tile.append(i)
            if up:
                tile_pos_x=level_tile_pos[i][0]
                tile_pos_y=level_tile_pos[i][1]
                if colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0],player_box[1]) \
                or colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0]+player_box_width,player_box[1]):
                    player_pos[1]+=move_speed
            elif down:
                tile_pos_x=level_tile_pos[i][0]
                tile_pos_y=level_tile_pos[i][1]
                if colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0],player_box[1]+player_box_height) \
                or colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0]+player_box_width,player_
box[1]+player_box_height):
                    player_pos[1]-=move_speed
            if right:
                tile_pos_x=level_tile_pos[i][0]
                tile_pos_y=level_tile_pos[i][1]
                if colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0]+player_box_width,player_box[1]) \
                or colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0]+player_box_width,player_
box[1]+player_box_height):
                    player_pos[0]-=move_speed
            elif left:
                tile_pos_x=level_tile_pos[i][0]
                tile_pos_y=level_tile_pos[i][1]
                if colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0],player_box[1]) \
                or colide(tile_pos_x,tile_pos_y,tile_width,tile_
height,player_box[0],player_box[1]+player_box_height):
                    player_pos[0]+=move_speed
            if gun.fire:
                if colide(level_tile_pos[i][0],level_tile_pos[i]
[1],tile_width,tile_height,
                gun.fire_pos[0]+gun.bullet.center[0],gun.fire_
pos[1]+gun.bullet.center[1]):
                    gun.fire_pos[0]-=gun.move_x
                    gun.fire_pos[1]-=gun.move_y
                    if not gun.death:
```

Ci sono zombie ovunque: meglio iniziare a sparare!

Uccidere non morti nei panni di un simil-Rambo è una delizia!

```
                        gun.moved_dist=gun.max_move_dist
```

Per aggiungere un ulteriore livello di profondità alla grafica del gioco potete disegnare delle ombre che, naturalmente, dovranno spostarsi nella stessa direzione del personaggio del giocatore:

```
    screen.blit(shadow,player_pos,(frame_pos_x,frame_
pos_y,cell_width,cell_height))
    screen.blit(sprite_sheet,player_pos,(frame_
pos_x,frame_pos_y,cell_width,cell_height))
```

Come sempre, in uno sparatutto il ruolo dei nemici è fondamentale. Dopo esservi assicurati che il loro codice sia stato correttamente implementato siete quasi alla fine del programma:

```
    for enemy in active_enemy:
        enemy.update(player_pos,gun,active_enemy,active_
solid_tile,level_tile_pos)
    active_enemy.sort(key=lambda x:x.pos[1])
    for enemy in active_enemy:
        if enemy.blit_ok:
            if enemy.pos[1]<=player_pos[1]:
                screen.blit(shadow,enemy.shadow_pos,enemy.
current_frame)
                screen.blit(enemy.image,enemy.pos,enemy.
current_frame)
                screen.blit(shadow,player_pos,(frame_
pos_x,frame_pos_y,cell_width,cell_height))
                screen.blit(sprite_sheet,player_pos,(frame_
pos_x,frame_pos_y,cell_width,cell_height))
            elif enemy.pos[1]>player_pos[1]:
                screen.blit(shadow,enemy.shadow_pos,enemy.
current_frame)
                screen.blit(enemy.image,enemy.pos,enemy.
current_frame)
            if enemy.glowing:
                screen.blit(enemy.glow,enemy.pos,enemy.
current_frame)
                enemy.glowing=0
```

Per poter sparare ai vostri nemici avete bisogno che i proiettili siano particelle funzionanti:

```
    if gun.fire:
        screen.blit(gun.bullet.image,gun.fire_pos,gun.
bullet.current_frame)
        if not gun.death:
            screen.blit(gun.bullet.shadow,gun.fire_pos,gun.
bullet.current_frame)
```

Infine, mostrate lo schermo con `pygame.display.flip()`. Premete **F5** in Python IDLE per salvare il codice, eseguirlo e iniziare a dare la caccia ai terribili morti viventi che infestano il monitor! LXP

L’eco dei LUG

I LUG

I LUG rappresentano da sempre il punto di riferimento per chiunque voglia conoscere GNU/Linux. Ogni mese dedicheremo loro questo spazio per la comunicazione di nuovi progetti e appuntamenti. Se hai qualcosa da segnalarci scrivi a **ecodeilug@linuxpro.it**

ABRUZZO

AnxaLUG - Lanciano
www.anxalug.org

Il Pinguino - Teramo
Non disponibile

OpenLUG - L’Aquila
Non disponibile

Pescara LUG
www.pescaralug.org

Pollinux LUG - Pollutri
Non disponibile

SSVLUG - San Salvo, Vasto, Termoli
www.ssvlug.org

TeateLUG - Chieti
Non disponibile

TeLUG - Teramo
www.telug.it

User Group Valle Roveto
http://linuxvalley-os4.blogspot.com/

BASILICATA

Basilicata LUG - Potenza e Matera
www.baslug.org

PLUG
www.pignolalug.it

CALABRIA

Bit01
www.associazionebit01.it

Bogomips - Bisignano
www.blug.it

CSLUG - Cosenza
http://cslug.linux.it

CzLug
Non disponibile

HackLab Catanzaro
http://hacklab.cz

HackLab Cosenza
https://hlcs.it

Piana LUG - Piana di Gioia Tauro
Non disponibile

SpixLug - Spezzano Albanese
Non disponibile

Verdebinario
www.verdebinario.org

CAMPANIA

GLUS
www.liberarete.it/glus

IGLUG - Napoli e provincia
www.iglug.org

IRLUG - Irpinia
www.irlug.it

LUG-Ischia
www.lug-ischia.org

LUG Acropoli
www.linux.it/~ciccios/lugagropoli.htm

Neapolis Hacklab
www.officina99.org/hacklab.html

EMILIA ROMAGNA

Borgotaro LUG - Val Taro
http://btlug.it/

ERLUG
http://erlug.linux.it

Ferrara LUG
www.ferrara.linux.it

FoLUG - Forlì
http://folug.linux.it

ImoLUG - Imola
www.imolug.org

PANLUG - Vignola
Non disponibile

PLUG - Parma
http://parma.linux.it

RELug - Reggio Emilia e provincia
http://relug.linux.it

RiminiLug
www.riminilug.it

S.P.R.I.Te
http://sprite.csr.unibo.it

UIELinux - Valle del Rubicone
www.uielinux.org

FRIULI VENEZIA GIULIA

GOLUG - Gorizia
www.golug.it

LUG Pordenone
www.pnlug.it

LugTrieste
http://trieste.linux.it

LAZIO

GioveLUG - Terracina
www.giovelug.org

Latina LUG
www.llg.it

LUG Privernum Volsca - Priverno (LT)
Non disponibile

LUG Rieti
lugrieti@linux.it

LUGRoma 3
www.lugroma3.org

TorLUG - Università Tor Vergata - Roma
http://lug.uniroma2.it/

LIGURIA

GE.P LUG - Genova
https://geplug.altervista.org/

Genova LUG
www.genovalug.altervista.org

Govonis GNU/LUG - Savona
www.govonis.org

TLug-TSL - Tigullio Ligure
http://tlug.linux.it/

LOMBARDIA

BGLug - Bergamo e provincia
www.bglug.it

BGLug Valle Seriana - Valle Seriana
http://bglugvs.web3king.com/

BrigX - Monza e Brianza
http://brigx.it

GL-Como - Como
www.gl-como.it

GLUX - Lecco e provincia
www.lecco.linux.it

GULLP - Gruppo Utenti Linux Lonate Pozzolo
www.gullp.it

HackLabCormano
http://hacklabcormano.it

LIFO - Varese
www.lifolab.org

LIFOS - Cinisello Balsamo
www.lifos.org

Linux Var - Varese
www.linuxvar.it

Lug8 - Gottolengo e Bassa Bresciana
http://lugotto.linux.it

LugBS - Brescia e provincia
http://lugbs.linux.it/

Lug Castegnato - Castegnato
www.kenparker.eu/LugCastegnato

LUG Legnano
https://luglegnano.wordpress.com

LugMan - Mantova e provincia
www.lugman.org

@italia_news

L'eco dei LUG

LugOB - Cologne e ovest bresciano
www.lugob.org
Lugotto - Gottolengo (BS)
http://lugotto.linux.it
POuL - Milano
www.poul.org
TiLug - Pavia
http://pavia.linux.it
VIMELUG - Linux User Group Vimercate
http://vimelug.org

MARCHE

CMlug
Non disponibile
FanoLUG
www.fanolug.org
GLM - Macerata
Non disponibile
PDP Free Software
https://pdp.linux.it

MOLISE

FrenterLUG - Larino
Non disponibile

PIEMONTE

BiLUG - Provincia di Biella
http://www.bilug.it
Gallug - Galliate
www.gallug.it
IvLug - Ivrea Linux User Group
www.ivlug.it
Linox Novara
www.linoxnovara.org

PUGLIA

BriLUG - Brindisi
Non disponibile
MurgiaLug - Santeramo in Colle
Non disponibile
ManfredoniaLug - Manfredonia
https://www.manfredonialug.it
SaLUG! - Salento
http://salug.it

SARDEGNA

GNUraghe - Oristano
www.gnuraghe.org
PLUGS - Sassari
Non disponibile

SICILIA

cLUG - Caltanissetta
Non disponibile
FreakNet MediaLab - Catania
www.freaknet.org
Free Circle
www.thefreecircle.org
Leonforte LUG
http://leonforte.linux.it
LUG Catania
www.catania.linux.it
LUGSR - Siracusa
www.siracusa.linux.it
MELUG - Messina
Non disponibile
Norp LUG - Noto, Pachino, Rosolini
Non disponibile
Poetry FreakNe
http://poetry.freaknet.org
VPLUG Linux Planet - Provincia Caltanisetta
www.vplug.it
SputniX - Palermo
www.sputnix.it

TOSCANA

ACROS - Area di Versilia, Lucca, Massa Carrara
www.lug-acros.org
Elbalinux
Non disponibile
ElsaGLUG - Val d'Elsa
www.elsaglug.org
FLUG - Firenze
www.firenze.linux.it
GOLEM - Empoli, Valdelsa
http://golem.linux.it
G.U.L.LI - Livorno
www.livorno.linux.it
GULP Pisa
www.gulp.linux.it
GuruAtWork - Grosseto e provincia
www.guruatwork.com
Lucca LUG
http://luccalug.it
L.U.G.A.R - Arezzo
Non disponibile
PtLug - Pistoia e provincia
www.ptlug.org

SLUG - Siena e provincia
www.siena.linux.it

TRENTINO ALTO ADIGE

LinuxTrent - Trento
http://linuxtrent.it
LugBz - Bolzano
www.lugbz.org

UMBRIA

OrvietoLUG
www.orvietolug.it
LUG Perugia
www.perugiagnulug.org

VALLE D'AOSTA

SLAG - Aosta
Non disponibile

VENETO

0421ug - Provincia di Venezia
www.0421ug.org
BLUG - Belluno
http://belluno.linux.it
GrappaLUG - Bassano del Grappa
https://grappalug.org/
LegnagoLUG
Non disponibile
Linux Ludus - Villafranca (VR)
www.linuxludus.it
LugAnegA
www.luganega.org
MontelLUG - Montebelluna
www.montellug.it
FSUG Padova
www.fsugpadova.org
TVLUG - Treviso
www.tvlug.it
VELug - Venezia
www.velug.it
AViLUG Schio
Non disponibile

NAZIONALI

Gentoo Channel Italia
www.gechi.it

@italia_news

In edicola dal 10 dicembre

NEL PROSSIMO NUMERO

Hai un argomento da proporci? Scrivi a **redazione@linuxpro.it**

Fatti la VPN!

Fra siti che non permettono l'accesso dall'Italia e la tua preziosa privacy, accedere a Internet con una rete virtuale è oggi essenziale: ecco come fare

Bimestrale - prezzo di copertina 5,90 €
www.linuxpro.it
redazione@linuxpro.it

Business Unit Manager: Massimiliano Zagaglia

Traduzione e localizzazione a cura di: **Dehive Sas di Andrea Orchesi**

Sprea S.p.A.
Sede Legale: Via Torino, 51 20063 Cernusco Sul Naviglio (Mi) - Italia
PI 12770820152- Iscrizione camera Commercio 00746350149
Per informazioni, potete contattarci allo 02 924321

CDA: Luca Sprea (Presidente), Alessandro Agnoli (Amministratore Delegato), Maverick Greissing (Consigliere Delegato)

ADVERTISING, SPECIAL PROJECTS & EVENTS
Marketing & Business Development: Marco Fregonara - Tel 347 2240482
marcofregonara@sprea.it
Segreteria: Emanuela Mapelli - Tel. 02 92432244 - emanuelamapelli@sprea.it
Events Production: Giulia Sprea (Mind Srl) - giulia@sprea.it

SERVIZIO QUALITÀ EDICOLANTI E DL
Sonia Lancellotti, Virgilio Cofano : Tel. 02 92432295
distribuzione@sprea.it - 351 5582739

ABBONAMENTI E ARRETRATI
Abbonamenti: si sottoscrivono on-line su www.sprea.it/linuxpro
abbonamenti@sprea.it
Tel 02 87168197 (lun-ven / 9:00-13:00 e 14:00-18:00)
Il prezzo dell'abbonamento è calcolato in modo etico perché sia un servizio utile e non in concorrenza sleale con la distribuzione in edicola.
Arretrati: si acquistano on-line su www.sprea.it/arretrati
abbonamenti@sprea.it
Tel 02 87168197 (lun-ven / 9:00-13:00 e 14:00-18:00)
329 3922420

FOREIGN RIGHTS
Gabriella Re: Tel . 02 92432262 - gabriellare@sprea.it

SERVIZI CENTRALIZZATI
Art director: Silvia Taietti
Grafici: Alessandro Bisquola, Nicole Bombelli, Tamara Bombelli, Nicolò Digiuni, Marcella Gavinelli, Luca Patrian
Coordinamento: Federica Berzioli, Chiara Civilla, Laura Galimberti, Michela Lampronti, Gabriella Re, Tiziana Rosato, Milena Sacchi, Roberta Tempesta, Silvia Vitali
Amministrazione: Irene Citino, Erika Colombo (responsabile), Désirée Conti, Sara Palestra- amministrazione@sprea.it
Ufficio Legale: Francesca Sigismondi

Linux Pro, pubblicazione registrata al Tribunale di Milano il 08.02.2003 con il numero 74. ISSN: 1722-6163

Direttore responsabile: Luca Sprea

Distributore per l'Italia: Press-Di Distribuzione stampa e multimedia s.r.l. 20090 Segrate
Distributore per l'Estero : SO.DI.P S.p.A. Via Bettola, 18 - 20092 Cinisello Balsamo (MI)
Tel. +390266030400 - Fax +390266030269 - sies@sodip.it - www.sodip.it
Stampa: Arti Grafiche Boccia S.p.A. Via Tiberio Claudio Felice, 7- 84131 Salerno

Copyright : Sprea S.p.A. - Future Publishing Limited

Informativa su diritti e privacy
La Sprea S.p.A. è titolare esclusiva della testata Linux Pro e di tutti i diritti di pubblicazione e di diffusione in Italia. L'utilizzo da parte di terzi di testi, fotografie e disegni, anche parziale, è vietato. L'Editore si dichiara pienamente disponibile a valutare - e se del caso regolare - le eventuali spettanze di terzi per la pubblicazione di immagini di cui non sia stato eventualmente possibile reperire la fonte. Informativa e Consenso in materia di trattamento dei dati personali (Codice Privacy d.lgs. 196/03). Nel vigore del D.Lgs 196/03 il Titolare del trattamento dei dati personali, ex art. 28 D.Lgs. 196/03, è Sprea S.p.A. (di seguito anche "Sprea"), con sede legale in Via Torino, 51 Cernusco sul Naviglio (MI). La stessa La informa che i Suoi dati, eventualmente da Lei trasmessi alla Sprea, verranno raccolti, trattati e conservati nel rispetto del decreto legislativo ora enunciato anche per attività connesse all'azienda. La avvisiamo, inoltre, che i Suoi dati potranno essere comunicati e/o trattati (sempre nel rispetto della legge), anche all'estero, da società e/o persone che prestano servizi in favore della Sprea. In ogni momento Lei potrà chiedere la modifica, la correzione e/o la cancellazione dei Suoi dati ovvero esercitare tutti i diritti previsti dagli artt. 7 e ss. del D.Lgs. 196/03 mediante comunicazione scritta alla Sprea e/o direttamente al personale Incaricato preposto al trattamento dei dati. La lettura della presente informativa deve intendersi quale presa visione dell'Informativa ex art. 13 D.Lgs. 196/03 e l'invio dei Suoi dati personali alla Sprea varrà quale consenso espresso al trattamento dei dati personali secondo quanto sopra specificato. L'invio di materiale (testi, fotografie, disegni, etc.) alla Sprea S.p.A. deve intendersi quale espressa autorizzazione alla loro libera utilizzazione da parte di Sprea S.p.A. Per qualsiasi fine e a titolo gratuito, e comunque, a titolo di esempio, alla pubblicazione gratuita su qualsiasi supporto cartaceo e non, su qualsiasi pubblicazione (anche non della Sprea S.p.A.), in qualsiasi canale di vendita e Paese del mondo.

Il materiale inviato alla redazione non potrà essere restituito.

FUTURE

Contenuti su licenza: Gli articoli in questa pubblicazione sono riprodotti o tradotti da Linux Format di cui i diritti e la licenza appartengono a Future Publishing Limited, di Future plc group company, UK 2021. Tutti i diritti sono riservati.

Linux Pro, è il marchio dato in licenza da Future Publishing Limited.

IN EDICOLA

OGNI 27 DEL MESE

★CLASSIC★ ROCK

OTTOBRE 2021

30 ANNI DI

PEARL JAM

STONEMUSIC.IT

STYLE
DRIGO (NEGRITA)
«TI DISEGNO IL BLUES»

THE ROLLING STONES
E ORA?

JUDAS PRIEST
METAL GODS

TUTTE LE NOVITÀ DEL MESE

- JOHN MELLENCAMP!
- SANTANA!
- THE PINEAPPLE THIEF!
- MARIANNE FAITHFULL!
- YES!
- MY MORNING JACKET!
- NICK CAVE & THE BAD SEEDS!
- SAMANTHA FISH!
- STURGILL SIMPSON!
- MUDDY WATERS!
- SHERYL CROW!
- POND!
- OSANNA!
- INGLORIOUS!

& MANY MORE

Sprea EDITORI

Acquistala su www.sprea.it/classicrock
versione digitale disponibile ogni 24 del mese